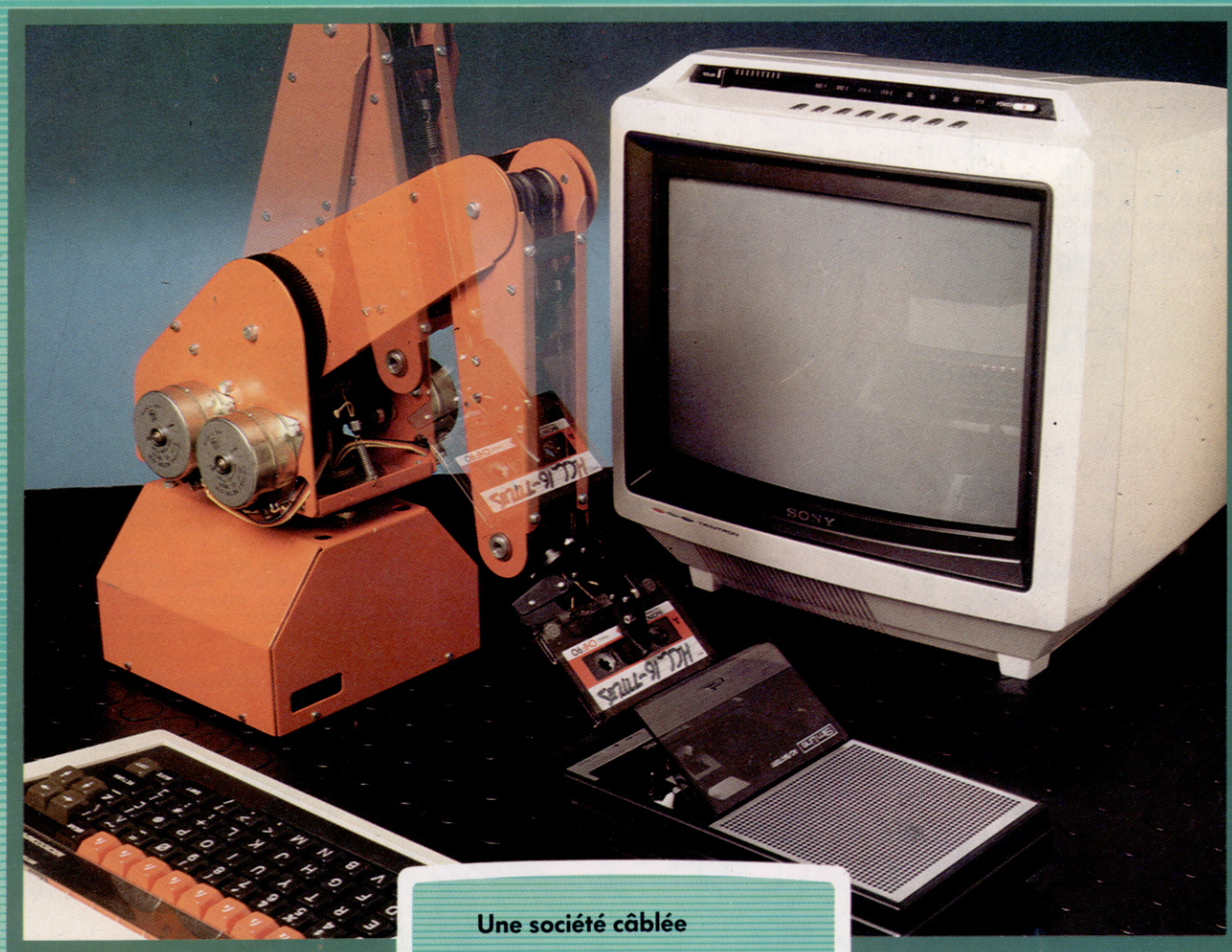


ABC

N° 16

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Une société câblée

Les mémoires du passé

Le Sharp MZ-711

Programmez votre robot

EDITIONS
ATLAS

M 6062-16-12F

85FB-3,80FS- \$ 1.95

HISTOIRE DE LA RÉVOLUTION FRANÇAISE

EN BANDES DESSINÉES

UNE SUPER PRODUCTION DES ÉDITIONS ATLAS

Mise en images par Ruffieux et Goeptert.

Écrite et réalisée par : Castellar et Laurence.

Conseiller historique pour les décors et les costumes : Carmignani.

Avec la participation des meilleurs spécialistes universitaires.

Pour la première fois, vous pourrez suivre

image par image, la RÉVOLUTION

FRANÇAISE, comme un grand

documentaire vivant et

passionnant. Voici, enfin dans

vos bibliothèques, la

reconstitution la plus fidèle

des événements qui ont fait

basculer l'ancien régime.

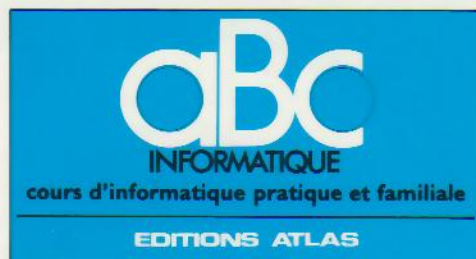
EDITIONS
ATLAS

**A LIRE
ABSOLUMENT!**

Une reconstitution historique
saisissante de vérité.

12 parutions-documents à collectionner.

Chaque semaine chez tous
les marchands de journaux.



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Luce. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume II, n° 16.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), J.-P. Bourcier (coordination), St.-André Laroche (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction).
Crédit photographique, couverture : Photo Orbis.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : avril 1984, 20844. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.
© Éditions Atlas, Paris, 1984.

A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



Société câblée

Le développement de la télédistribution par câble offrira un nouveau moyen de communication aux utilisateurs d'ordinateurs domestiques.



Pour la majorité du grand public, la télévision par câble est à la télévision classique par voie hertzienne ce que l'hypermarché est à la petite épicerie de quartier : une possibilité de multiplication extraordinaire du nombre des chaînes mises à la disposition du possesseur d'un poste de télévision. Il est d'ailleurs parfaitement exact que le câble permettra bientôt, du moins techniquement, de recevoir quinze, vingt, voire trente canaux de télévision. Mais on oublie encore trop souvent, lorsqu'on parle de télédistribution par câble, toutes les autres possibilités de services télématiques et informatiques qui pourront ainsi être mis à la disposition des usagers, et qui devraient radicalement bouleverser, au-delà de nos habitudes de téléspectateurs, l'essentiel de nos modes de vie et de nos façons de communiquer avec l'extérieur.

Tout d'abord, un bref rappel historique des principales étapes de la télévision par câble s'impose. Au commencement était le cuivre... Entendez par là que, depuis déjà plusieurs décennies, on sait comment acheminer, autrement que par le biais des ondes hertziennes, des images et des sons jusqu'à un poste de télévision : pour cela, on les fait transiter, sous forme d'impulsions électriques codées, dans un câble

électrique en cuivre dit « câble coaxial ». Les réseaux, dont la structure se ramifie depuis l'émetteur jusqu'aux usagers, sont de forme « arborescente ». Mais l'usager demeure, dans la plupart des cas, le récepteur passif des programmes qui sont diffusés à son intention. En France, ce type de réseau existait déjà dans quelques petites zones géographiques très circonscrites et appelées « zones d'ombre », car leur configuration topographique (vallée encaissée, comme à Munster par exemple) ne leur permettait pas de recevoir les faisceaux hertziens dans de bonnes conditions. Le câble faisait donc figure de palliatif. Dans les années soixante-dix, on câble également, à titre d'expérience pilote, une douzaine de villes dites « nouvelles » (Cergy-Pontoise, Évry, Marne-la-Vallée, L'Isle-d'Abeau, etc.). Mais ces réseaux ne servent encore, dans leur très grande majorité, qu'à véhiculer les trois chaînes traditionnelles de la télévision française puisqu'ils sont toujours soumis au monopole de programmation de la puissance publique qui régit l'audio-visuel français.

Au tout début des années quatre-vingt, la technique de la télédistribution fait un bond en avant, avec l'invention de la fibre optique et la

Cité câblée

Aux États-Unis, on installe automatiquement le câble lors du développement de nouveaux quartiers. Chaque foyer dispose ainsi de dizaines de chaînes de télévision et de nombreuses stations de radio. Il serait même possible de composer un réseau intégré d'information qui permettrait de mesurer de façon centrale la consommation d'électricité et de gaz. De plus, tous les immeubles publics nouvellement construits possèdent un réseau local permettant ainsi aux ordinateurs de l'immeuble de communiquer entre eux et avec des ordinateurs situés à l'extérieur. (Cl. Ian McKinnell.)



Station satellite

Pour la première fois en France, il est possible de regarder les émissions d'une chaîne de télévision dont les images sont diffusées par le biais d'un satellite. Mais pour avoir cette « chance », il vous faut prendre une chambre à l'hôtel Méridien de Paris. En effet, depuis le 1^{er} août 1983, cet hôtel capte des émissions de la chaîne de télévision britannique Satellite Television. Les signaux, reçus sur une antenne parabolique de 3 m de diamètre placée sur le toit de l'hôtel, sont décryptés à l'aide d'un équipement spécial fourni par les P.T.T. et mis au standar Secam. (Cl. hôtel Méridien, Paris.)

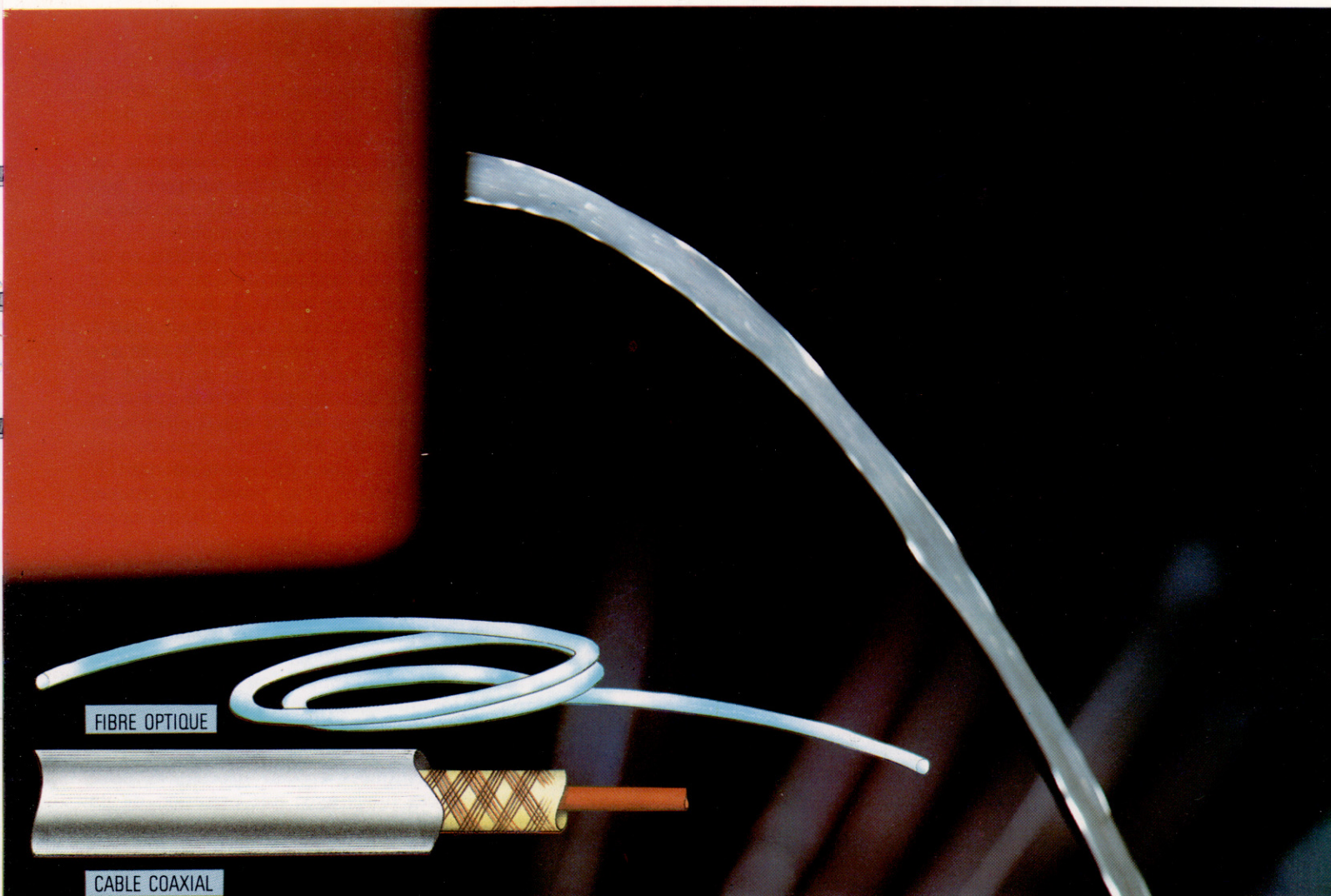
mise au point, en laboratoire, de réseaux à structure étoilée. Le principe est très simple : remplacer le traditionnel câble de cuivre, coûteux et encombrant, par une fibre de silicium fine et souple comme un cheveu. On ne module plus désormais du courant électrique mais des rayons lumineux, de manière un peu analogue aux jets d'eau d'une fontaine éclairée canalisant la lumière sur leur parcours. Pour transmettre des impulsions lumineuses de ce type, il faut disposer d'une source lumineuse très intense et très modulable (ce qui a été rendu possible grâce à la maîtrise du laser), et pouvoir « filer » à volonté des cheveux de verre d'une transparence parfaite. Or, on sait maintenant produire en quantité industrielle de telles fibres optiques (on étire

sur une distance de plusieurs kilomètres une barre de silice de 1 m de longueur, sous certaines conditions de température, de pression et de mélanges gazeux). Le procédé demande encore à être perfectionné puisqu'il entraîne une quantité trop importante de déchets. Il reste enfin à régler à l'échelle de la production industrielle tous les problèmes de « connectique », c'est-à-dire l'ensemble des procédés, mariant l'optique et l'électronique de pointe, qui permettent d'interconnecter les fibres entre elles et de constituer ainsi les fameux réseaux en étoile. Chaque fibre partant de la prise d'un usager remonte jusqu'au cœur de l'étoile où elle peut soit capter des canaux de télévision ou des lignes téléphoniques (ou être reliée à des services audiovisuels à la carte : médiathèques, vidéothèques et autres téléthèques), soit être branchée sur des services télématiques (vidéotex, téléalarme, télésurveillance, banques de données...), soit être directement interconnectée avec la fibre d'un autre usager (visiophone, télex...).

Outre ses indéniables avantages matériels sur le câble coaxial (encombrement cent fois inférieur ; 1 g de fibre optique remplace, à performance égale, 10 kg de cuivre importé), la fibre optique possède également des potentialités techniques infiniment supérieures : cible passive d'une diffusion d'images et de sons à sens unique, le téléspectateur traditionnel va devenir, grâce à elle, un élément actif dans la gigantesque sarabande optoélectronique joliment baptisée du nom d'« interactivité ».

Pour un usager possesseur d'un simple récepteur de télévision branché en fibre optique et d'un clavier semblable à ceux qui équipent les micro-ordinateurs domestiques, l'interactivité signifie donc la possibilité concrète de dialoguer, par écrans interposés, avec toutes les banques de données reliées au réseau : annuaire électronique, réservation d'hôtel, de billets de train ou d'avion, consultation des programmes municipaux de spectacles, cinémas ou expositions, gestion de ses comptes bancaires à distance, commandes informatisées chez ses fournisseurs habituels. Autant de services mis à la disposition de l'homme de demain sans qu'il ait besoin de quitter son domicile. Dans la mesure où l'on n'a pas encore inventorié toutes les possibilités d'application de cette nouvelle forme de télédistribution, sur le plan des services notamment, il est très difficile de prévoir dès maintenant les conséquences probables d'une telle révolution des moyens de communication sur les comportements et les modes de vie des générations à venir. Certains sociologues se sont naturellement penchés sur la question. Mais leur champ d'investigation semble s'être essentiellement limité à l'émergence probable de micro-communautés fortement tournées vers la recherche de nouveaux types de convivialité. Cette hypothèse, relativement optimiste, n'est pas dénuée d'arrière-pensées idéologiques et culturelles. Elle présente l'inconvénient majeur de négliger les risques d'auto-enfermement et de coupure face au monde réel que fera inévitable-





FIBRE OPTIQUE

CABLE COAXIAL

Câble ou fibre ?

Les fibres optiques, développées d'abord en 1966, utilisent le phénomène de la réflexion interne totale. La lumière, introduite à l'une des extrémités, circule dans les fibres optiques avec une perte de luminosité très faible. Elle est réfléchie chaque fois qu'elle rencontre la paroi externe, tout comme elle le serait dans un prisme ou dans un jet d'eau. Pour obtenir ce résultat, le niveau de pureté du verre doit approcher les 100 %. On utilise une lumière infrarouge produite par un laser car la fréquence des impulsions lumineuses doit être très rapide.

Les câbles coaxiaux, utilisés pour connecter les antennes des téléviseurs, sont composés d'un fil de cuivre assez gros, entouré d'un réseau de fils plus fins.

(Cl. Gary Marsh.)

ment courir à l'homme de l'an 2000 l'« hypermédiatisation » de son environnement. Pourquoi ne pas imaginer, en effet, que nous puissions bientôt effectuer la quasi-totalité de nos activités sociales, professionnelles ou de consommateurs, confortablement installés dans notre living, les yeux rivés en permanence sur un poste de télévision transformé en écran de visualisation et de contrôle de tous les actes de notre vie quotidienne? La question mérite d'être posée même si de nombreux problèmes sont loin d'être réglés. Problèmes techniques, d'abord : le passage du laboratoire à la production industrielle (et à son corollaire l'exploitation commerciale) nécessite que soit encore considérablement améliorée toute l'infrastructure électronique permettant la connexion des fibres en réseaux. Mais aussi, et surtout, problèmes politiques et juridiques liés à la diffusion généralisée de réseaux câblés sur l'ensemble du paysage urbain français. (Le câble est par essence une technologie de télédistribution en milieu urbain, les campagnes nécessitant pour leur part un balayage plus large, du type de celui que peuvent prendre en charge des satellites de télévision directe, ou satellites DBS [Direct Broadcasting System].) Un « plan câble » a été

lancé depuis plusieurs mois en France. Il prévoit le câblage en fibres optiques de six millions de foyers d'ici à 1992, sous la responsabilité de la Direction générale des télécommunications.



Une image vaut-elle mille mots ?

Il est difficile de déterminer le nombre de bits nécessaires pour composer une image télévisée en couleurs. Mais si l'on retient qu'une largeur de bande de 9 MHz permet une transmission à 4,5 millions de bauds, et que chaque image est transmise 25 fois par seconde, un balayage est composé de 180 000 bits. Mille mots en nécessitent 60 000.



Mémoire du passé

Le développement de dispositifs de stockage d'information plus compacts est l'objectif poursuivi par les concepteurs de systèmes informatiques.

Chaque ordinateur doit pouvoir stocker de l'information. Dans la plus simple opération, comme l'addition, un chiffre doit être stocké ou « retenu » jusqu'à ce qu'il soit utilisé dans l'étape suivante du calcul. Depuis la naissance de l'ordinateur, les capacités des mémoires n'ont cessé de croître pour résoudre des problèmes de plus en plus complexes.

L'électricité est le support essentiel de stockage de l'information. On peut imaginer l'électricité comme une circulation d'électrons ou comme un déplacement ondulatoire. Mais, dans ces deux représentations, la caractéristique fondamentale est le mouvement. L'impossibilité d'immobiliser quelque chose qui, par sa nature, reste toujours en mouvement, a mené à l'adoption de méthodes de stockage indirectes. Une batterie, par exemple, stocke de l'énergie sous une forme chimique; dans une centrale hydro-électrique, l'énergie potentielle stockée dans une masse d'eau est délivrée aux turbines. Nous allons voir comment la manipulation des données a été résolue en faisant appel aux signaux électriques.

Au cours de la Seconde Guerre mondiale, des recherches intensives furent menées pour séparer le signal réfléchi par un avion en mouvement

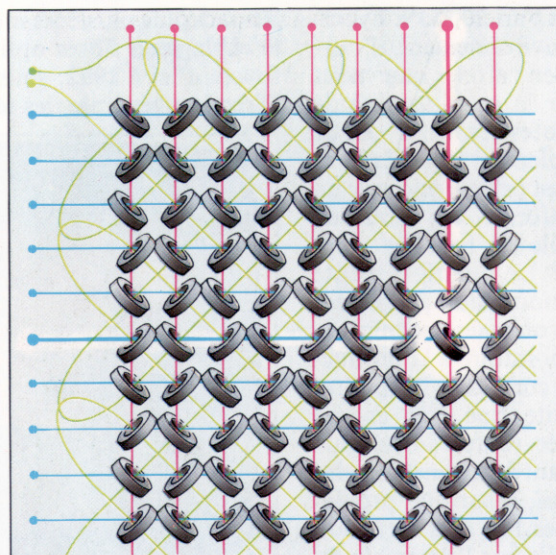
et les impulsions provenant d'objets immobiles comme des arbres. Un dispositif nommé « ligne de retard au mercure » fut alors inventé. Ce dispositif pouvait stocker temporairement les impulsions d'ondes radio et les comparer aux ondes réfléchies lors de balayages successifs de façon à éliminer les contours immobiles.

Une ligne de retard au mercure consiste en un tube de verre d'une longueur de 1 m, rempli de mercure et muni d'un quartz à chaque extrémité. Lorsqu'un signal électrique est appliqué à l'une des extrémités, le quartz vibre et crée une onde qui est transmise à l'autre extrémité du tube où elle est détectée par l'autre quartz et convertie de nouveau en un signal électrique. L'onde met environ 660 microsecondes (μs) pour traverser le tube. En retransmettant le signal dans le mercure, il est possible de stocker les impulsions pendant plusieurs minutes avant qu'elles ne soient brouillées.

Avec une horloge synchronisant le train d'impulsions à 500 000 cycles par seconde, 330 bits pouvaient être stockés dans un tube de 1 m. La taille de la mémoire pouvait être augmentée en allongeant le tube, mais en ralentissant le temps d'accès, puisqu'une impulsion ne pouvait être lue que lorsqu'elle se trouvait dans le circuit électrique à l'extérieur du tube. Ces systèmes étaient volumineux et très coûteux. Des techniques sophistiquées étaient nécessaires pour construire ces tubes avec une précision de un centième de millimètre.

Une autre méthode de stockage de données fut inventée par F.C. Williams (université de Manchester), qui se pencha sur l'utilisation de l'électricité statique produite sur la surface interne d'un tube à rayon cathodique comme moyen de stockage. Un canon électronique forme une configuration de charges statiques sur l'écran, et cette configuration est détectée par un réseau de fils placé contre la surface extérieure de l'écran. En 1947, il était possible de stocker 2 048 bits d'informations sur un seul écran pendant des heures. Bien que le temps d'accès fût rapide, les charges statiques devaient être régénérées toutes les 30 μs , sinon elles étaient perdues.

L'utilisation de bandes magnétiques fut tentée avec le système LEO en Grande-Bretagne, puis sur l'UNIVAC (*UNIVersal Automatic Computer*) aux États-Unis à la fin des années quarante. Ce fut la première technique fiable qui permit de stocker économiquement de grandes quantités d'information. Huit bits étaient stockés sur une colonne dans la largeur de la bande magnétique avec une densité de 2 360 colonnes par centimètre. Avec une longueur totale de 61 m ou plus, il était possible d'obtenir des mémoires permanentes de 1 méga-octet. Cependant, même avec les moteurs d'entraînement les plus rapides, le temps d'accès aux données situées en milieu de bande était de quelques secondes. Ce type de stockage ne fut utilisé que pour stocker



Mémoires à tores

Tout anneau peut être adressé individuellement en appliquant un courant dans les fils horizontaux et verticaux

appropriés. Le fil de lecture capte les variations du flux magnétique des divers anneaux, et déduit ainsi s'ils renferment des 1 ou des 0. (Cl. Kevin Jones.)



des fichiers où les données étaient appelées séquentiellement, comme pour préparer la paie des employés d'une entreprise. La bande fut ensuite accélérée en insérant un coussin d'air entre la tête de lecture et la bande, ce qui eut pour effet de réduire l'usure due à la friction.

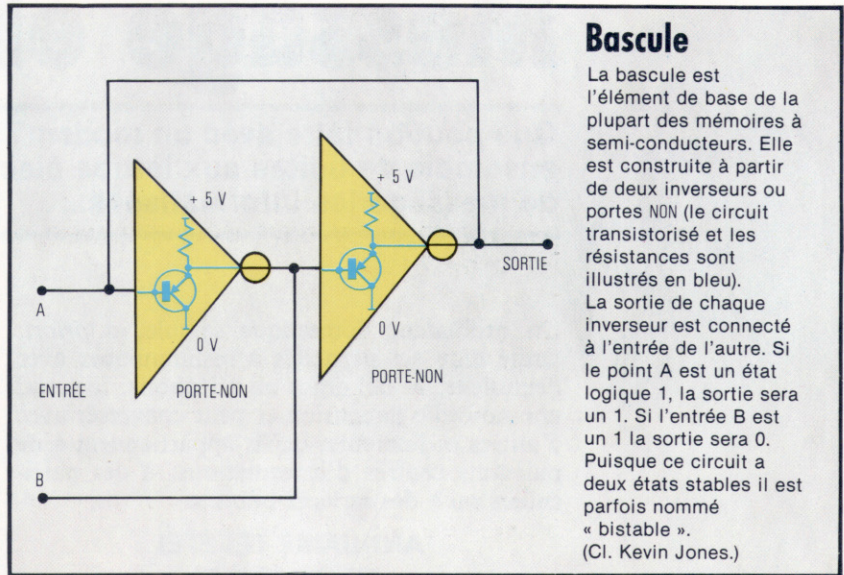
Une mémoire d'une grande capacité, d'un temps d'accès très rapide et d'un coût réduit de stockage par bit fut inventée au Massachusetts Institute of Technology en 1950. La mémoire à tores utilisait des anneaux métalliques enfilés sur un réseau de fils. Un courant passant dans un fil qui traverse un anneau métallique crée un champ magnétique sur l'anneau. Ce champ magnétique peut être orienté dans deux directions qui servent à représenter les deux états binaires, 0 et 1. La direction du champ magnétique de l'anneau peut être inversée par un courant approprié. Comme un courant produit un champ magnétique, l'inverse est aussi vrai : lorsque la direction du champ magnétique est inversée, une petite quantité de courant est induite. Le champ magnétique n'est inversé que lorsque le courant dépasse un seuil. En n'appliquant que la moitié du courant de seuil dans un fil vertical et l'autre moitié dans un fil horizontal, seul un tore du treillis recevra assez de courant pour être inversé.

Le diamètre des anneaux fut réduit de 4 mm à quelques centièmes de millimètre. La puissance nécessaire pour écrire des données dans une cellule fut d'autant diminuée. L'information était lue dans une cellule particulière de mémoire en tentant de l'inverser. En surveillant le courant induit, il était possible de découvrir si la cellule avait été inversée ou non et de déduire ainsi son état initial. Cependant, cette méthode de lecture créait un même état magnétique dans toutes les cellules. Il était donc nécessaire d'écrire de nouveau les données après chaque lecture.

Avec l'intégration à grande échelle des transistors sur une seule puce, l'une des premières méthodes de stockage de données — déjà utilisée dans l'ENIAC — réapparut : la « bascule ». Une bascule est un dispositif stable dans l'un de ses deux états (il forme un circuit dit « bistable »). Il est composé de deux interrupteurs électroniques mis dos à dos, où la sortie de l'un est transmise dans l'entrée de l'autre. De cette façon, une impulsion peut être emprisonnée dans la bascule jusqu'à ce qu'elle soit sollicitée.

Les puces RAM statiques, que l'on retrouvait dans les premiers ordinateurs domestiques, renferment des milliers de circuits bistables dont chacun stocke 1 bit. La plupart des ordinateurs récents utilisent une RAM dynamique. L'utilisateur ne remarque aucune différence, mais une RAM dynamique est plus rapide et moins chère à construire. Elle peut être comparée à un ensemble de condensateurs dont chacun peut stocker une charge électrique. Mais une charge stockée a toujours tendance à s'échapper. Le contenu de cette mémoire doit donc être « régénéré » des milliers de fois par seconde.

Les tout derniers types de mémoire qui viennent de faire leur apparition sont les mémoires à

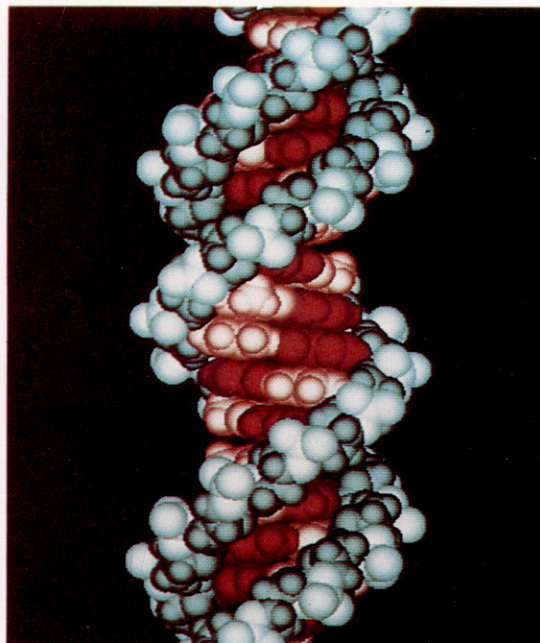


Bascule

La bascule est l'élément de base de la plupart des mémoires à semi-conducteurs. Elle est construite à partir de deux inverseurs ou portes NON (le circuit transistorisé et les résistances sont illustrés en bleu). La sortie de chaque inverseur est connecté à l'entrée de l'autre. Si le point A est un état logique 1, la sortie sera un 1. Si l'entrée B est un 1 la sortie sera 0. Puisque ce circuit a deux états stables il est parfois nommé « bistable ». (Cl. Kevin Jones.)

bulles et les mémoires optiques à laser. Ces dernières sont similaires aux disques à lecture laser utilisés pour stocker de la musique ou des films vidéo. Dans une mémoire à bulles, de petits champs magnétiques, ou bulles, sont créés dans une puce de matière magnétique. Ces bulles peuvent être déplacées dans des boucles et leur présence ou absence représente un 1 ou un 0 binaire. La densité potentielle de stockage est si élevée que ce type de mémoire est appelé à remplacer les systèmes actuels de stockage sur cassette et sur disquette, dans un avenir relativement proche.

Les mémoires cryogéniques deviendront peut-être elles aussi disponibles. A des températures voisines du zéro absolu, la résistance électrique d'un support devient presque nulle et on obtient la « superconductibilité ». Grâce à cette propriété, la vitesse et la capacité des systèmes futurs dépasseront, et de loin, ce que nous connaissons aujourd'hui.



Biotechnologie

Même si la micro-électronique a atteint un niveau incroyable de miniaturisation, chaque opération dans une mémoire à semi-conducteur implique la circulation de milliers d'électrons. Les scientifiques essaient déjà de concevoir des mémoires qui fonctionnent au niveau moléculaire. Mais à ce stade nous parlons plutôt de biotechnologie que de micro-électronique. On sait par exemple que la structure d'une molécule d'ADN renferme le code génétique qui commande la croissance et l'entretien des cellules de notre corps. Nous ne pourrions cependant pas reproduire cette forme de stockage avant de nombreuses années.

Message électronique

Que peut-on faire avec un modem? Connecter son ordinateur à un ensemble de boîtes aux lettres électroniques et former un réseau de messageries informatisées...

Un ordinateur domestique semble, *a priori*, limité dans ses capacités à communiquer avec l'extérieur. Mais, grâce au téléphone, il prend une nouvelle dimension et peut converser avec d'autres ordinateurs, qu'ils appartiennent à de puissants centres d'informations, à des particuliers ou à des services publics.

ANNUAIRE TÉLÉTEL

(quelques exemples)

Services administratifs		
Assistel	service permanent	(3) 614.91.66
Melcom	de 9 h à 19 h	(41) 88.98.29
La Poste	service permanent	(3) 613.91.55
TSD Télématique	service permanent	(91) 48.84.40
Télem Nantes	de 10 h à 20 h	(40) 35.75.00

Services commerciaux		
Cirtel	service permanent	(56) 34.45.18 ou (56) 34.09.23
C.I.S.I.	horaires variables selon les services	(3) 613.91.55
Eurodial	de 8 h à 24 h	(3) 613.91.55 ou (3) 604.91.65
Octel	service permanent	(3) 613.91.55
Questel	de 8 h à 23 h	(3) 613.91.55
Prestintel		(3) 613.91.55

Le réseau télématique qu'installent en France les pouvoirs publics s'appelle Télétel. Par son intermédiaire, il est possible d'échanger des messages avec d'autres utilisateurs d'ordinateurs ou de terminaux du type Minitel, passer des commandes à certains fournisseurs ou encore échanger des programmes. Il n'existe pas de contrainte particulière pour offrir des services quelconques sur le réseau Télétel, si ce n'est de posséder un ordinateur muni d'une importante capacité mémoire. Les distances ne posent pas de problème au niveau des communications. Vous pouvez appeler un correspondant dans votre propre ville, à l'autre bout du pays ou même à l'étranger. La seule restriction viendra éventuellement de votre note de téléphone.

Mais, avant de vouloir faire « parler » votre ordinateur familial au téléphone, il est nécessaire de l'équiper d'un matériel supplémentaire : le modem. Souvent, il existe une entrée « modem » à l'arrière de votre ordinateur. S'il n'en possède pas (c'est le cas du Vic-20 et du Spectrum par exemple), vous devez vous procurer une carte d'interface série que vous adapterez à votre matériel. Suivant les cas, le modem est directement connecté à la ligne du téléphone

par l'intermédiaire d'une prise supplémentaire ou par un coupleur acoustique. Les modems utilisés sur le réseau Télétel permettent des vitesses de transmission des informations de 1 200/75 bauds. Ce rapport signifie que les informations qui partent du terminal vers l'ordinateur central sont acheminées à la vitesse de 75 bauds et que, en retour, la vitesse est de 1 200 bauds. Cette remarque n'est pas sans importance. La plupart des ordinateurs ne sont pas adaptés à ces vitesses. Thomson sort une interface de branchement à Télétel sur son TO 7. Il vous faut donc acheter un modem si vous voulez vous brancher. Les prix sont encore élevés en France, entre 4 000 et 6 000 F. Mais on peut espérer que, comme en Grande-Bretagne, les prix vont chuter.

Il ne suffit pas de posséder un modem pour avoir accès aux services Télétel. Votre ordinateur doit disposer d'un logiciel lui permettant de communiquer dans les normes vidéotex qui sont appliquées en France (types et nombres de caractères affichables sur l'écran, procédures conversationnelles, etc.). Certains constructeurs en proposent sur leur modèle, comme l'Apple II, le TI-99 ou le TO 7. Mais rien ne vous empêche de réaliser votre propre programme pour faire de votre ordinateur un terminal Télétel. Si vous disposez déjà d'un ordinateur familial et que vous trouvez trop cher ou trop compliqué de le transformer en terminal Télétel, il vous est possible de louer le petit terminal Minitel que propose l'administration (pour 70 ou 120 F par mois) et d'utiliser sa prise péri-informatique pour y connecter votre ordinateur.

Imaginons que vos problèmes techniques soient résolus, c'est-à-dire que le modem et le logiciel de votre ordinateur soient bien compatibles avec le réseau. La première chose que vous devez entreprendre est de composer le numéro de téléphone du correspondant que vous voulez joindre. Mais à ce niveau existe une double possibilité : vous composez soit l'un des trois numéros « standard » du service Télétel, soit directement celui de votre correspondant. Pourquoi ces deux options? C'est une question de tarification. En ligne directe, vous aurez à payer selon les mêmes principes que pour le téléphone (durée et distance). En passant par l'un des standards, vous ne paierez que suivant le type de services que vous avez demandé, et, la plupart du temps, la distance n'entre plus en jeu car la communication s'établit sur des réseaux de communication spécialisés (réseau Transpac en France, par exemple).

TÉLÉTEL

Missive

La poste électronique Missive est un système de messageries électroniques développé en France. Les utilisateurs louent une boîte aux lettres dans la mémoire d'un ordinateur. Chaque abonné peut envoyer un message à un autre, qui peut le lire et le copier sur son propre ordinateur. Des connexions existent entre Missive et d'autres réseaux internationaux. L'incompatibilité entre les systèmes est ainsi effacée au niveau des connexions.



Services de messages informatisés

L'une des applications les plus utiles d'un modem est l'accès à un réseau (un tableau d'affichage électronique). Des messages peuvent être « épinglés » ou lus par toute personne ayant le mot de passe adéquat. Il est possible d'annoncer diverses rencontres ou encore de proposer des équipements d'occasion. Il est même possible de transférer sur cassette ou sur disquette des jeux ou d'autres listings de programme. (Cl. SIC-PTT.)

La première chose qui apparaît sur l'écran, une fois la communication établie, est une sorte d'image de bienvenue, que l'on appelle « page d'accueil ». Elle correspondra à celle du service que vous avez appelé si vous êtes passé par sa ligne directe. Si vous êtes passé par le standard Télétel, vous devez passer aux pages écran qui suivent la page accueil par une instruction « suite ». Pour cela, vous devrez taper le code d'accès du service ou du correspondant demandé. Le choix est double : vous tapez soit le nom abrégé du service qui vous intéresse, soit son numéro d'identification. Ces informations sont disponibles dans le nouvel annuaire des services Télétel publié par les P.T.T.

L'accès aux informations d'un service prend alors plusieurs directions. Dans la première, le service vous demande de remplir un formulaire où figurent, par exemple, le lieu désiré de vos vacances, le montant de la location que vous souhaitez et la période. En retour, une page écran vous donne les réponses. Une deuxième méthode consiste à vous proposer des choix successifs (recherche arborescente) : le service présente un menu et par étapes (de menu en menu) vous arrivez à l'information recherchée. Le troisième système d'accès est plus direct, car il se fait par mot clé. Par exemple, si vous désirez aller en Bretagne et que vous interrogez une agence de voyages, il vous suffira de taper BR pour signifier le lieu choisi.

Au cas où vous seriez perdu dans vos manipulations, le service Télétel met toujours à votre disposition une instruction « guide » pour vous aider dans vos recherches.

Mais il ne faudrait pas croire qu'en consultant l'annuaire des services télématiques qui sont proposés en France il vous est possible d'avoir accès à tous les centres qui proposent de l'information. En effet, certains services sont publics (informations administratives ou commerciales), donc ouverts à tous, et d'autres sont strictement privés, réservés aux abonnés. Dans

ce dernier cas, l'accès au service nécessite la connaissance d'un code et d'un mot de passe. La porte d'entrée ne s'ouvre pas à tous les coups !

Rien ne vous empêche de votre côté d'utiliser votre ordinateur familial comme « serveur », c'est-à-dire comme support de prestations de services sur le réseau de communication. Il vous sera néanmoins nécessaire de disposer d'une importante capacité de mémoire et d'adapter votre ordinateur aux normes Télétel en mettant en place un programme « interface » adéquat. Une des utilisations les plus intéressantes du système Télétel concerne les messageries électroniques. De quoi s'agit-il ? Dans la mémoire d'un ordinateur central, plusieurs abonnés disposent d'un espace qui correspond à une sorte de boîte aux lettres. Chaque boîte est adressée. Les abonnés d'un service de messageries peuvent envoyer à partir de leur propre ordinateur un message à l'adresse « informatique » du correspondant désiré. Mais seul le propriétaire de la boîte peut « entrer » à l'intérieur de son adresse pour prendre connaissance des messages reçus. Pour cela, il dispose d'un code confidentiel qu'il crée lui-même. Ces services de messageries se développent beaucoup à l'intérieur des entreprises. Plus question de perdre du temps à attendre, au téléphone, que votre correspondant soit là ou que la communication devienne libre. En outre, dans le cas de messageries internationales, le problème des fuseaux horaires disparaît puisque vous pouvez envoyer ou lire un message vingt-quatre heures sur vingt-quatre.

Rapidement, les messageries électroniques vont concurrencer le service classique des postes, et le téléphone qui devient trop encombré. Une filiale de la Direction générale des télécommunications, la société France Câbles et Radio, propose Missive aux usagers. Il ne fait aucun doute que, dans un avenir très proche, les micro-ordinateurs vont devenir les outils privilégiés des nouvelles communications.

Éditeurs

La plupart des micro-ordinateurs permettent d'éditer des programmes à l'écran, ce qui fait gagner beaucoup de temps.

Il arrive à tout le monde de faire des erreurs en utilisant un clavier d'ordinateur. C'est pourquoi il est nécessaire de disposer de fonctions d'édition. Dans de nombreuses situations, il est nécessaire de modifier les données affichées : corriger une faute de frappe, modifier une instruction erronée, ou mettre à jour une information qui a été modifiée.

Plusieurs programmes d'applications comportent certaines formes d'« éditeurs » spécialisés. Les fonctions d'édition d'un programme de traitement de texte, par exemple, sont conçues pour modifier la réalisation de documents. Ces fonctions permettent d'effacer des phrases, de déplacer des paragraphes entiers à d'autres endroits dans le document, de remplacer des mots ou des expressions par d'autres.

La plupart des ordinateurs domestiques ont un éditeur intégré dans leur système d'exploitation. Il est destiné principalement à éditer les listages de programmes. Mais, dans tout programme, les erreurs de syntaxe surviennent régulièrement, et, très souvent, des problèmes apparaissent lors de l'exécution. Ceux-ci doivent être résolus. Les fonctions offertes par l'éditeur de votre ordinateur peuvent faire toute la différence lors du développement d'un long programme. Nous devons cependant insister sur le fait qu'un bon programmeur doit consacrer beaucoup de temps à tester le fonctionnement de son programme sur papier avant d'entrer le listage dans l'ordinateur. Il n'est pas du tout recommandé de taper la première solution qui vous vient à l'esprit, et donc de passer les neuf dixièmes du temps de développement du programme à sa mise au point!

ordinateurs domestiques. Les éditeurs de ligne s'inspirent de la période où tout le travail sur ordinateur était réalisé à partir de télétypes ou de terminaux éloignés de l'ordinateur. Les télétypes n'avaient qu'un tampon de 80 caractères (80 octets). Le programmeur pouvait obtenir une liste imprimée du programme entier en tapant LIST, mais s'il désirait modifier la ligne 120 par exemple, il devait la retaper complètement. Sur certains systèmes, taper EDIT 120 affichait cette ligne particulière, et les modifications ou effacements pouvaient être faits à l'aide des touches de retour arrière ou d'effacement (les insertions étaient encore impossibles). D'autres commandes comme DELETE (avec des numéros de ligne spécifiés) furent ajoutées, mais il était toujours nécessaire d'appeler et de modifier une ligne à la fois.

Les éditeurs de nombreux ordinateurs domestiques fonctionnent toujours comme s'ils ne possédaient qu'un tampon d'une ligne, alors que l'écran est stocké complètement en mémoire — chaque emplacement de caractère à l'écran correspond à 1 octet de mémoire.

Un éditeur d'écran est beaucoup plus efficace. Il vous permet de déplacer facilement des textes et des graphiques. Lorsque vous appuyez sur la touche RETURN, l'éditeur lit la ligne sur laquelle se trouve le curseur et l'envoie à l'interpréteur, où elle est exécutée (si la ligne est une commande) ou entrée dans le programme (si elle commence par un numéro de ligne). A l'aide des quatre touches fléchées, l'utilisateur peut placer le curseur sur tout point du programme affiché et modifier ainsi les caractères à volonté.

Un éditeur d'écran doit être écrit en code machine afin d'être assez rapide et comporter certaines fonctions extrêmement utiles. Les meilleurs éditeurs d'écran permettent de faire défiler un listage vers le haut comme vers le bas, d'insérer ou d'effacer des lignes complètes et des caractères individuels. Certains offrent même des commandes similaires à celles qui sont offertes par des programmes de traitement

Ligne par ligne

L'éditeur du Sinclair Spectrum est bien meilleur que la plupart des éditeurs de ligne, bien qu'il ne soit vraiment pas aussi facile à utiliser qu'un éditeur d'écran. Pour changer une ligne particulière, l'indicateur de position (>), qui apparaît entre le numéro de ligne et la ligne elle-même, doit être déplacé sur la ligne désirée à l'aide des touches de déplacement.

```

5 REM first draft
6 DIM a(12)
10 FOR i=1 TO 100
20 READ a
30 NEXT i
40 PRINT "which value"
50 INPUT v
60 LET t=a(v)*1.15
70 PRINT t
80 FOR j=1 TO 100
90 LET r=r+(j)
95 NEXT j
100 PRINT "total is ";r
100 DATA 12,24,7.5,90
101 DATA 30,21.5,76,95
102 DATA 12,5,43,0.9
110 STOP
  
```

```

5 REM first draft
6 DIM a(100)
10 FOR i=1 TO 100
20 READ a
30 NEXT i
40 PRINT "which value"
50 INPUT v
60 LET t=a(v)*1.15
70 PRINT t
80 FOR j=1 TO 100
90 LET r=r+(j)
95 NEXT j
100 PRINT "total is ";r
100 DATA 12,24,7.5,90
101 DATA 30,21.5,76,95
102 DATA 12,5,43,0.9
110 STOP
  
```

Il existe deux types d'éditeurs : les *éditeurs d'écran* et les *éditeurs de ligne*. Les premiers sont beaucoup plus souples et plus faciles à utiliser. Les derniers sont plus répandus sur les

de texte, comme la recherche et le remplacement de toutes les apparitions d'une chaîne de caractères. Les éditeurs deviennent de plus en plus sophistiqués et faciles à utiliser.



Le Sharp MZ-711

Un ordinateur domestique peu coûteux dont certains des périphériques optionnels sont intégrés à même le boîtier.

Le Sharp MZ-711 est un ordinateur intéressant qui présente des caractéristiques inhabituelles. Sa construction correspond aux normes élevées de conception japonaise. L'affichage couleur est excellent — très stable et d'une bonne brillance — et, à l'exception d'un manque de netteté lorsque certaines combinaisons de couleurs sont utilisées, les caractères sont très lisibles.

Deux aspects de sa conception sont cependant assez curieux. Le clavier, par exemple, serait très agréable si sa disposition n'était pas exactement l'inverse de la normale : les majuscules sont sélectionnées par défaut, tandis que les minuscules sont obtenues en pressant les touches SHIFT. Bien que très pratique en programmation BASIC, cette disposition du clavier rend toute autre utilisation extrêmement difficile. En plus, il semble n'y avoir aucune méthode pour convertir le clavier à la normale, malgré ce que dit le manuel.

L'unité à cassette du Sharp MZ-711 est un dispositif offert en option. Elle est encastrée

dans le boîtier principal, mais la plupart des utilisateurs choisissent de l'inclure dans la machine lors de son acquisition. Ce système est lent mais fiable et le chargement semble toujours s'effectuer à la première tentative. Néanmoins, le fonctionnement a plusieurs points faibles. Il n'est possible d'utiliser que des noms de fichiers très rudimentaires, et le moteur ne peut être commandé — ce qui aurait facilement pu être prévu. L'efficacité de l'unité diminue considérablement. Ces limites sont surprenantes, d'autant que Sharp est expert dans d'autres domaines, particulièrement dans l'industrie de la haute-fidélité, où ses lecteurs de cassettes commandés par solénoïde sont courants.

L'imprimante-traceur intégrée (de nouveau un périphérique optionnel encastré dans le boîtier) possède un mécanisme d'impression et des marqueurs délicats et très vulnérables. Ces marqueurs doivent être retirés de la machine et bouchés après utilisation. Malheureusement, ils sont très petits et donc faciles à perdre lorsqu'ils



Clavier

Le clavier du Sharp MZ-711 donne une apparence professionnelle à la machine. Les touches sont correctement espacées, et permettent une frappe bien adaptée au traitement de texte. Cinq touches de fonction programmables, répondant aux touches SHIFT, offrent dix fonctions et facilitent l'utilisation de la machine. Les symboles graphiques sont disposés en séquence logique, alors que les autres touches — étant disposées dans le format QWERTY — ne le sont pas. Il peut donc être difficile de trouver quelles combinaisons de touches produisent les symboles désirés.

(Cl. Chris Stevens.)



sont retirés de l'imprimante. Ils ont également tendance à sécher rapidement, et la qualité de l'impression laisse souvent à désirer.

L'interpréteur BASIC est chargé à partir de la bande au lieu d'être stocké en ROM. Cela permet d'utiliser la machine avec d'autres langages.

La documentation technique est excellente. Elle comporte des topographies mémoire et des schémas de montage, un listage en assembleur du logiciel intégré et d'autres détails techniques. Un manque : elle ne comprend pas de schéma de synchronisation, mais elle est meilleure que la plupart des manuels offerts par d'autres constructeurs d'ordinateurs domestiques.

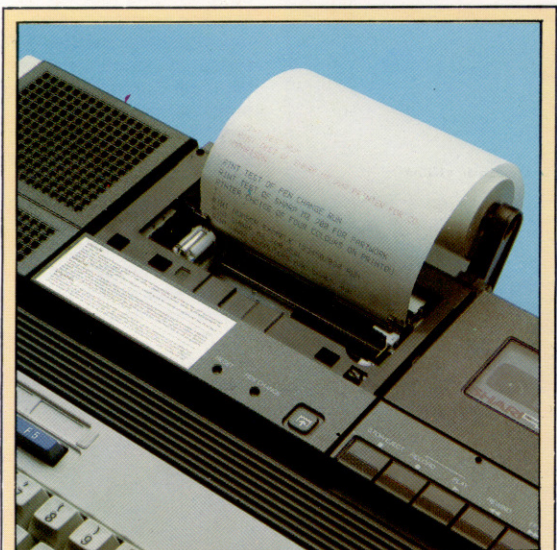
Les fabricants du Sharp MZ-711 ont une tendance inexplicable à minimiser certaines de ses caractéristiques. Le programme de démonstration en est un bon aperçu. Par exemple, le manuel indique que la machine ne peut produire que les huit couleurs habituelles, mais le programme semble réussir au moins huit fois plus de couleurs.

Cependant, en inspectant le programme, on peut remarquer qu'une courte routine en code machine a été écrite à l'adresse 40960 et qu'elle est appelée à plusieurs endroits du programme pour produire divers demi-tons de couleur. Le nombre réel de couleurs est donc difficile à établir et la documentation est timide.



Unité à cassette

L'unité à cassette est en option, bien qu'elle puisse être intégrée dans le boîtier. Cependant, le système d'exploitation ne peut pas commander le moteur de cette unité, ce qui limite ses possibilités.



Imprimante/traceur

Tout comme la cassette, l'imprimante/traceur est un dispositif complémentaire offert en option. Trois modes de fonctionnement permettent trois dimensions de texte, soit 26, 40 ou 80 colonnes de largeur.

Connecteur d'imprimante

Commutateur d'imprimante

Ce bouton permet de sélectionner une imprimante interne ou externe.

Connecteur d'alimentation

Connecteur du haut-parleur

ROM

Ce système ROM ne contient qu'un très court programme nommé moniteur qui sert à manipuler des emplacements de mémoire ou à charger l'interpréteur BASIC à partir de la cassette.

RAM

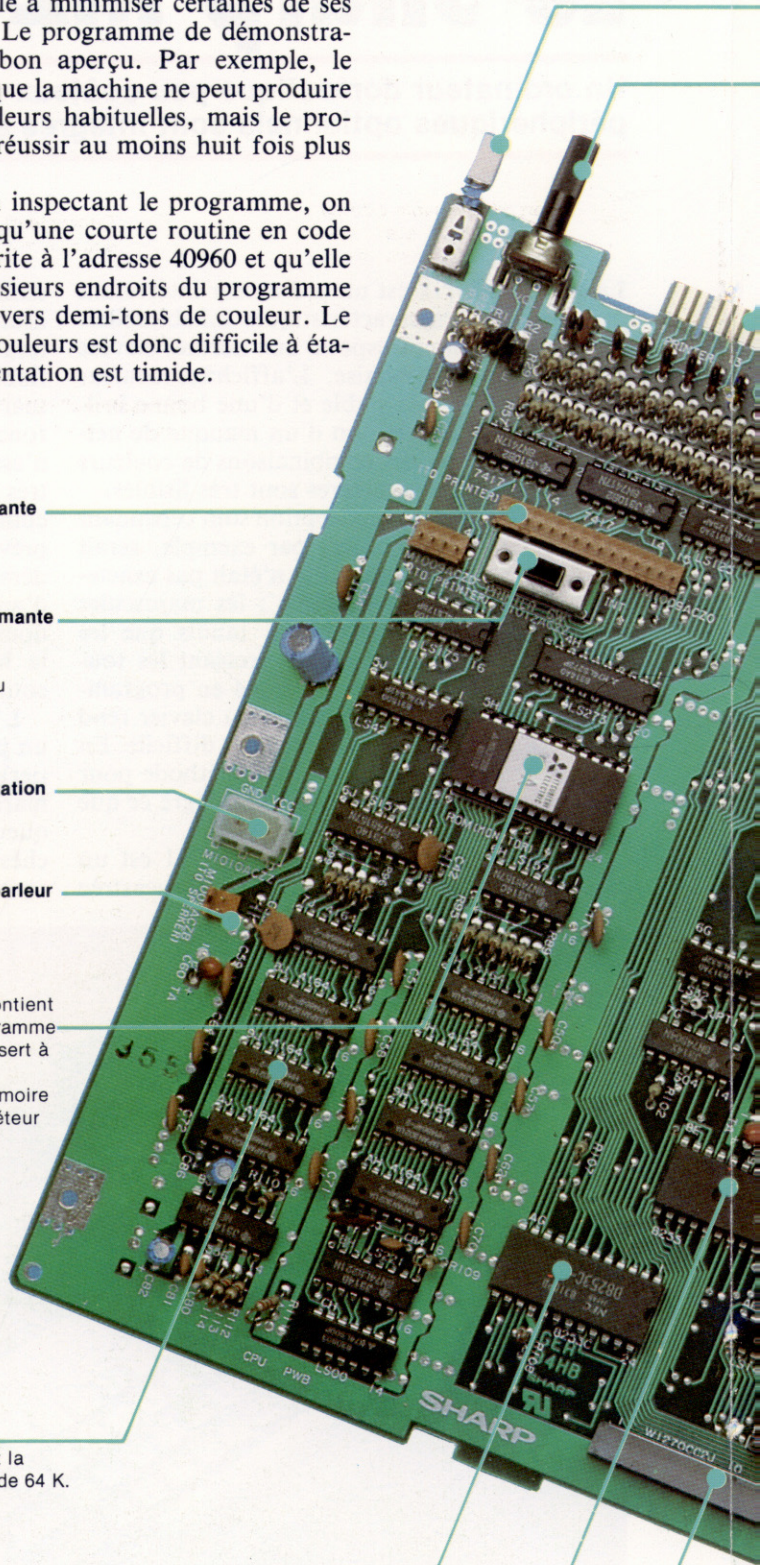
Huit puces forment la mémoire standard de 64 K.

Horloge

Adaptateur d'interface de périphérique

Cette puce commande l'interface avec le clavier et l'unité à cassette.

Connecteur du clavier





SHARP MZ-711

PRIX

***, unité à cassette*, imprimante/traceur*.

DIMENSIONS

440 x 305 x 86 mm.

UC

Z80A

VITESSE DE L'HORLOGE

4 MHz.

MÉMOIRE

ROM 12 K, RAM 64 K

AFFICHAGE VIDÉO

24 lignes de 40 caractères, ou graphiques basse résolution 50 x 80 points, 8 couleurs officielles et un nombre non spécifié de nuances, l'arrière-plan et le premier plan peuvent être définis indépendamment; 127 caractères prédéfinis et 127 caractères définis par l'utilisateur.

INTERFACES

RS232 série, cassette, bus système, imprimante parallèle.

LANGAGE FOURNI

BASIC sur cassette.

AUTRES LANGAGES DISPONIBLES

PASCAL, ASSEMBLEUR, FORTRAN, FORTH, DAFMOD, TOOLKIT

ACCESSOIRES FOURNIS

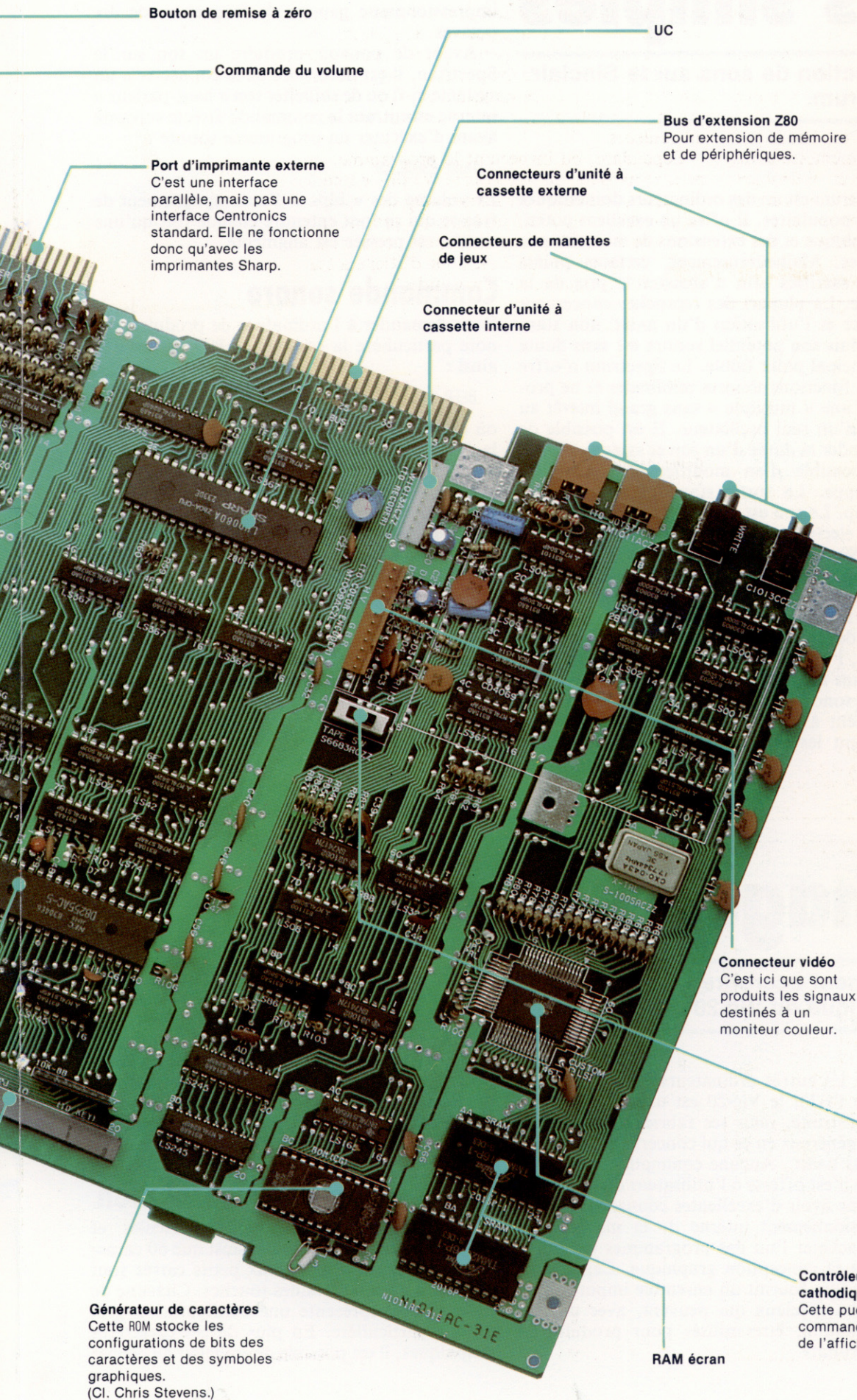
Notices d'installation et manuel BASIC, fil de téléviseur et de cassette, cassette BASIC et de démonstration.

CLAVIER

Touches mécaniques et cinq touches de fonction programmables répondant aux touches SHIFT.

DOCUMENTATION

Très complète, elle traite d'importants détails techniques. Elle est plutôt ennuyeuse mais fournit une très bonne introduction à la machine.



Bouton de remise à zéro

Commande du volume

Port d'imprimante externe

C'est une interface parallèle, mais pas une interface Centronics standard. Elle ne fonctionne donc qu'avec les imprimantes Sharp.

UC

Bus d'extension Z80

Pour extension de mémoire et de périphériques.

Connecteurs d'unité à cassette externe

Connecteurs de manettes de jeux

Connecteur d'unité à cassette interne

Connecteur vidéo

C'est ici que sont produits les signaux destinés à un moniteur couleur.

Commutateur de lecture cassette

Si votre unité à cassette ne réussit pas à charger les programmes, ce commutateur modifie la phase du signal et corrige habituellement le problème.

Contrôleur de tube à rayons cathodiques

Cette puce faite sur mesure commande tous les aspects de l'affichage vidéo.

Générateur de caractères

Cette ROM stocke les configurations de bits des caractères et des symboles graphiques.
(Cl. Chris Stevens.)

RAM écran



Sons simples

Production de sons sur le Sinclair Spectrum.

Le Spectrum est un des ordinateurs domestiques les plus populaires. Il offre un excellent potentiel graphique et des extensions de mémoire peu coûteuses. Malheureusement, certains points ont été sacrifiés afin d'abaisser le prix de la machine. La plupart des reproches concernent le clavier et l'utilisation d'un BASIC non standard. Mais son potentiel sonore est sans doute son principal point faible. Le Spectrum n'offre que des fonctions sonores minimales et ne produit qu'une « musique » sans grand intérêt au moyen d'un seul oscillateur. Il est possible de commander la durée d'un son et sa note, mais il est impossible d'en modifier le registre ou l'enveloppe. La sortie standard est un autre handicap. Le son est produit par un petit haut-parleur électrique et n'est guère évolué. Cependant, Sinclair a prévu d'autres sorties de signaux à partir des ports cassette « écoute » ou « micro » qui permettent une amplification externe au moyen d'un système hi-fi. Mais, en raison de la mauvaise qualité des fonctions sonores, l'intérêt est réduit. Le Spectrum offre cependant l'avantage de la simplicité des commandes sonores BASIC, comme BEEP et PAUSE, qui permettent à l'utilisateur de comprendre plus facilement les principes de génération sonore

sur ordinateur. Et la machine possède une impressionnante gamme de fréquences de dix octaves.

Avant de pouvoir produire un son sur le Spectrum, il est nécessaire de le connecter à un système hi-fi ou de solliciter son « haut-parleur » interne en entrant la commande directe suivante avant d'exécuter un programme sonore :

POKE 23609,100

Le volume des « bips » d'accompagnement de frappe qui se font entendre chaque fois qu'une touche est pressée est augmenté.

Commande sonore

Pour demander à l'ordinateur de produire une note particulière la commande BEEP est utilisée ainsi :

BEEP d,n

où « d » représente la durée et « n » la hauteur de la note. La durée peut être réglée à une valeur comprise entre 0,001 25 et 10 secondes; et la note est exprimée par un nombre de demi-tons à partir du *do* situé sous le *la* du diapason (une valeur de 0) allant de - 60 à 69. Par exemple, l'instruction suivante produit la note *la* à 440 Hz, qui est à neuf demi-tons du *do* de référence, pendant 0,5 seconde :

BEEP 5,9

Pour exécuter une chaîne de notes avec une pause précise entre chacune, nous pouvons utiliser la commande PAUSE :

PAUSE ms

Images primaires

Comment accroître le potentiel graphique du Vic-20

Comme les autres ordinateurs Commodore (le 64 et le PET), le Vic-20 est une machine très bien construite, mais ses fabricants n'ont pas été très généreux en ce qui concerne le jeu d'instructions BASIC. Aucune commande graphique spéciale n'est offerte à l'utilisateur du Vic-20. Il faut donc avoir d'excellentes connaissances sur le fonctionnement interne de la machine ou encore acheter l'un des programmes utilitaires d'aide à la conception graphique. Cependant, Commodore fournit un ensemble important de caractères spéciaux qui peuvent, avec un peu d'imagination, être utilisés pour produire de bons résultats.

Seize couleurs sont disponibles sur le Vic-20 et chaque emplacement de caractères peut afficher quatre couleurs différentes. L'affichage est composé de 23 lignes de 22 cellules de caractères, 8 points par 8 points, mais les caractères peuvent aussi être affichés dans un format rectangulaire de 16 points par 8. Un mode haute résolution est disponible mais il est plutôt difficile à programmer pour l'utilisateur moyen.

Potentiel en basse résolution

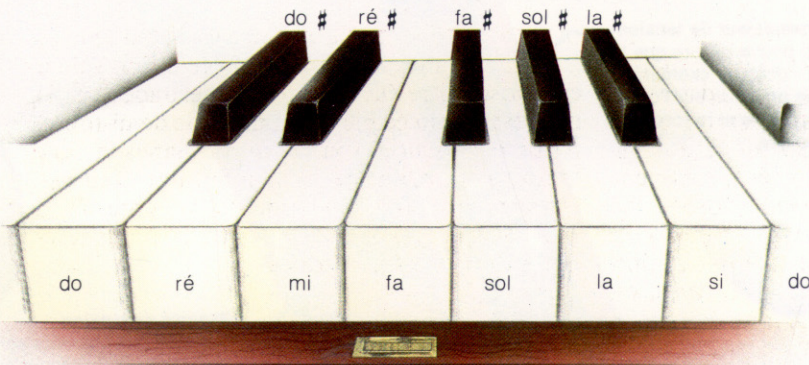
Les caractères alphabétiques minuscules et majuscules sont disponibles, ainsi que 60 caractères graphiques PET. Deux petits carrés sont imprimés devant certaines touches. Chacune de ces touches représente une configuration graphique particulière. En plus de ces caractères graphiques, il est possible de solliciter au clavier



où « ms » représente le temps en millisecondes. Pour exécuter une octave en *do* majeur (*do, ré, mi, fa, sol, la, si, do*) à partir du *do* sous le *la* du diapason, avec une pause de 0,5 seconde entre chaque note, le programme suivant peut être utilisé :

```
10 FOR I = 1 TO 8
20 READ N
30 BEEP .5,N
40 PAUSE 250
50 NEXT I
60 DATA 0, 2, 4, 5
70 DATA 7, 9, 11, 12
```

Ce programme produit une gamme. Une octave va d'une note d'origine (dans ce cas *do* sous le *la* du diapason) jusqu'à la note suivante portant le même nom (le *do* suivant) et couvre douze demi-tons. On nomme cela une octave (huit notes dans une gamme majeure).



Gary Marsh

DO majeur

des symboles de jeu de cartes, des damiers, des cercles et d'autres encore qui peuvent être assemblés à l'écran pour dessiner des graphiques, composer des tableaux, produire divers lettrages et créer d'autres effets. Chaque caractère peut être affiché inversé (noir sur blanc au lieu de blanc sur noir), ce qui accroît les possibilités graphiques.

Des caractères peuvent être affichés au moyen de l'instruction PRINT ou en écrivant (avec POKE) les codes appropriés dans les mémoires écran et couleur. La version Commodore de l'instruction PRINT est extrêmement puissante; elle permet à l'utilisateur de définir la couleur de chaque caractère individuellement. Les mouvements du curseur peuvent aussi être commandés à partir de l'instruction PRINT afin de produire facilement des affichages mobiles. L'instruction POKE, pour afficher des caractères, n'est pas aussi rapide que l'instruction PRINT, mais cette méthode peut être parfois utile.

Potentiel en haute résolution

Il est possible de créer des graphiques haute résolution, mais la taille de la mémoire ne permet d'utiliser que la moitié de l'écran. Le processus qui permet d'obtenir ce type de haute résolution s'appelle « manipulation de configuration binaire », une technique qui permet au

programmeur de commander chaque point dans une zone donnée de l'écran. Chaque cellule de caractère sur l'écran est formée de 64 points disposés en 8 lignes de 8 points. Une relation mathématique entre les coordonnées d'un point (x, y) et le bit correspondant dans la matrice de caractère existe. On peut donc l'utiliser, en combinaison avec les commandes POKE, pour produire des affichages point par point.

Cartouche Super Expander

Produire des graphiques haute résolution par manipulation de configuration binaire peut être un processus long et difficile. Une autre approche est possible en achetant la cartouche Super Expander qui permet de disposer de commandes BASIC pour produire un affichage couleur haute résolution, et de la musique. Parmi les commandes haute résolution, mentionnons GRAPHIC, qui sollicite le mode d'affichage, POINT pour définir des points sur l'écran, et PAINT.

L'utilisation de la cartouche présente deux inconvénients principaux : aucune commande UNPOINT pour effacer un point, et impossibilité de « peindre » les deux côtés d'une ligne diagonale sans faire disparaître la ligne elle-même.

Les graphiques basse résolution du Vic-20 sont très bien conçus et très souples. Les graphiques haute résolution sont difficiles à utiliser.

Gammes et pianos

En utilisant la commande INKEY\$, le clavier du Spectrum peut être utilisé pour simuler approximativement le clavier d'un piano :

```
10 REM *****
20 REM * OCTAVE PIANO *
30 REM *****
40 IF INKEY$ = « Q » THEN BEEP 1,0
50 IF INKEY$ = « 2 » THEN BEEP 1,1
60 IF INKEY$ = « W » THEN BEEP 1,2
70 IF INKEY$ = « 3 » THEN BEEP 1,3
80 IF INKEY$ = « E » THEN BEEP 1,4
90 IF INKEY$ = « R » THEN BEEP 1,5
100 IF INKEY$ = « 5 » THEN BEEP 1,6
110 IF INKEY$ = « T » THEN BEEP 1,7
120 IF INKEY$ = « 6 » THEN BEEP 1,8
130 IF INKEY$ = « Y » THEN BEEP 1,9
140 IF INKEY$ = « 7 » THEN BEEP 1,10
150 IF INKEY$ = « U » THEN BEEP 1,11
160 IF INKEY$ = « I » THEN BEEP 1,12
170 GOTO 40
```

Plusieurs améliorations peuvent être apportées à un programme comme celui-ci afin de créer un « instrument » plus efficace. Généralement, le Spectrum ne peut guère être utilisé comme source sonore en raison des sons médiocres qu'il produit. En revanche, il est un excellent outil d'apprentissage musical. La seule manière d'obtenir un son acceptable est d'acheter un matériel complémentaire compatible avec l'ordinateur.

```
100 REM ** VIC 20 RABBIT **
110 POKE36879,110
120 PRINT"
130 PRINT"
140 PRINT"
150 PRINT"
160 PRINT"
170 PRINT"
180 PRINT"
190 PRINT"
200 PRINT"
210 PRINT"
220 PRINT"
230 PRINT"
240 PRINT"
250 PRINT"
260 PRINT"
270 PRINT"
280 PRINT"
290 PRINT"
300 PRINT"
310 PRINT"
320 GOTO320
```

Exemple d'affichage sur Vic-20

Ce listing de programme pour le Vic-20 démontre les possibilités offertes par le jeu de caractères spéciaux du Commodore. Le dessin du lapin est créé entièrement à l'aide des caractères prédéfinis. Il est ainsi facile de voir exactement ce que donnera l'affichage final.



Haut les mains!

Les petits bras robots donnent un avant-goût de la programmation de commande. Ils peuvent être connectés à tout ordinateur muni d'un port parallèle.

Avez-vous déjà souhaité demander à votre ordinateur d'accomplir une tâche simple, comme faire une tasse de thé? Programmer votre ordinateur pour qu'il commande une cuisinière ne pose aucun problème avec une interface adéquate. Mais, lorsque vous désirez lui faire manipuler des objets, comme verser de l'eau chaude dans une théière, il est nécessaire de posséder un bras mécanique. De tels dispositifs — nommés bras robots — viennent récemment de faire leur apparition sur le marché. Ce sont des versions simplifiées des bras industriels utilisés par des constructeurs automobiles comme Renault ou Peugeot dans les travaux de soudure ou de peinture. Le bras robot Armdroid, de Colne Robotics, qui fut probablement le premier bras robot offert pour des ordinateurs domestiques, fit son apparition en 1981. Bien que le bras ne soit pas mobile (sauf si vous le montez sur un robot de sol), il permet de manipuler les objets avec une remarquable précision.

Pour établir une interface entre le bras et un ordinateur, un simple port parallèle à huit bits suffit. Un bit détermine si l'information est transmise vers le robot ou en provenance du robot. Trois bits d'adresse sont utilisés pour sélectionner le moteur désiré, et les quatre autres bits commandent la direction, l'amplitude et la vitesse du mouvement. Des signaux d'horloge sont aussi envoyés pour synchroniser les mouvements et les instructions de l'ordinateur. Pour accélérer le processus et permettre au

bras d'effectuer des manœuvres complexes, des bascules électroniques sont intégrées dans les circuits de commande.

Afin de commander au bras de se placer en position pour saisir un objet, il est d'abord nécessaire de diviser le mouvement global en une série d'éléments simples. Chaque moteur doit être commandé individuellement; les mouvements précis imprimés définissent le mouvement global du bras. L'information est alors stockée dans la mémoire de l'ordinateur et le

Bobines

Les bobines de transmission sont disposées de telle sorte que si l'angle de l'épaule est changé, l'angle du coude est automatiquement modifié afin que l'avant-bras demeure au même angle par rapport à l'horizontale.

Coude

Cette articulation a une liberté de mouvement de 270°.

Biceps

Avant-bras

Compensateur de tension

Cette poulie assure que les trois « doigts » exercent la même pression sur l'objet saisi, même si la forme est irrégulière.

Ressort de tension

Tous les mouvements sont transférés des moteurs au bras au moyen de ressorts, et ceux-ci doivent toujours être tendus pour garantir la précision du mécanisme.

Main

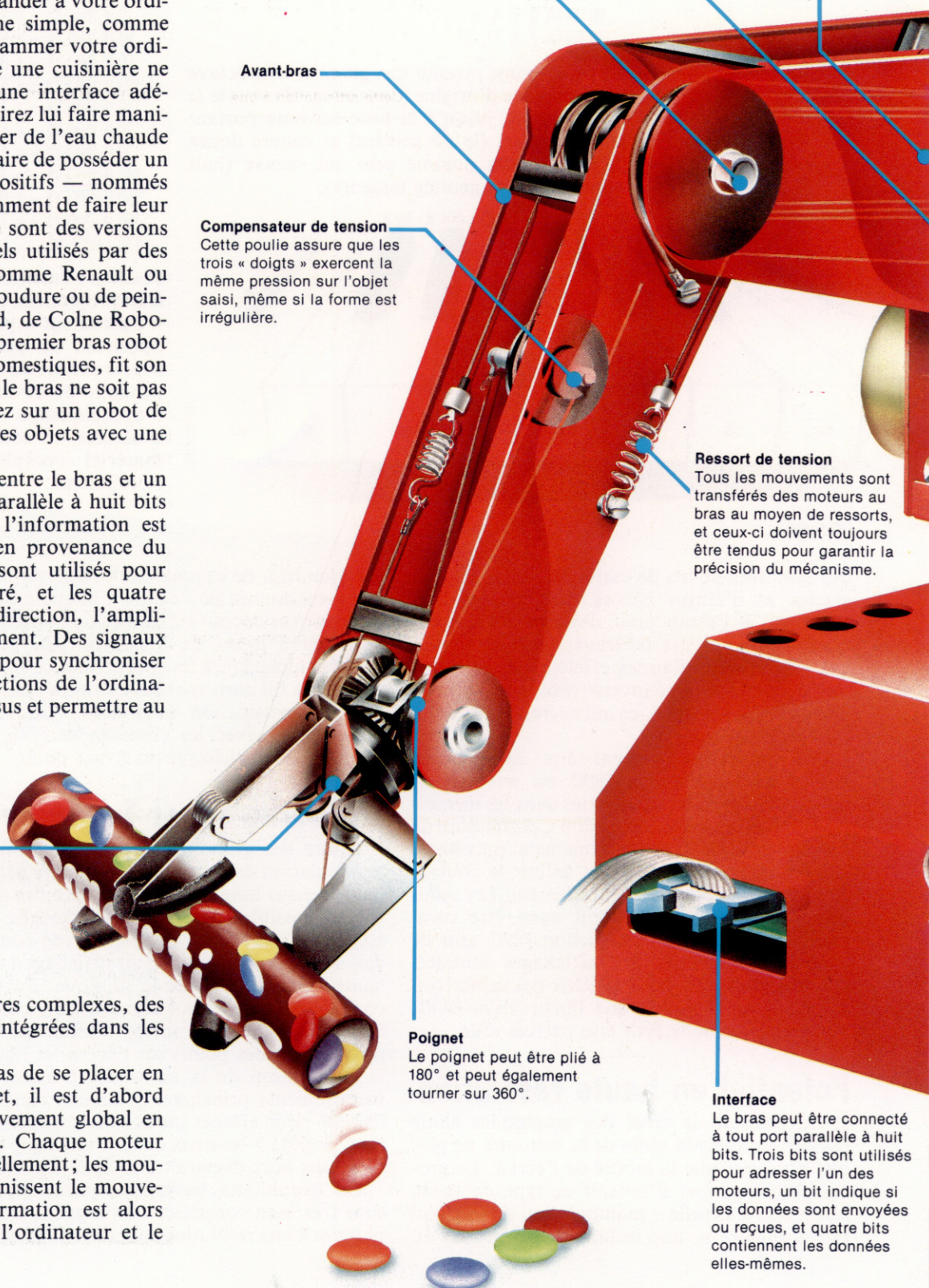
Les trois « doigts » de la main ont des articulations à ressort et des coussinets de caoutchouc pour aider à saisir les objets.

Poignet

Le poignet peut être plié à 180° et peut également tourner sur 360°.

Interface

Le bras peut être connecté à tout port parallèle à huit bits. Trois bits sont utilisés pour adresser l'un des moteurs, un bit indique si les données sont envoyées ou reçues, et quatre bits contiennent les données elles-mêmes.





mouvement peut être répété à volonté. La plupart des bras robots disponibles actuellement sont livrés avec des programmes qui comportent des routines servant à « apprendre » une séquence de mouvement.

Si le bras doit manipuler des objets délicats — le test classique est un œuf —, l'ordinateur doit être en mesure de contrôler la pression de la prise. Trop faible, l'œuf tombera; trop forte, elle risque de briser la coquille. Diverses méthodes sont employées pour transmettre l'information du bras à l'ordinateur, mais la plus répandue

implique de simples micro-interrupteurs. Ces dispositifs peuvent servir à définir la limite de la course du bras (les moins coûteux ne possèdent pas de capteurs), ou être intégrés dans la pince afin de détecter la limite de pression prédéterminée.

Un autre système de détection de pression (utilisé sur les bras plus sophistiqués) implique l'utilisation de certains matériaux dont la résistance électrique varie selon la pression. Ces variations de résistance peuvent être calculées, et la pression exercée sur le capteur est communiquée à l'ordinateur. Bien que cette méthode soit plus coûteuse, elle permet d'obtenir des résultats plus précis.

Si le programme ne prévoit aucune information en provenance du bras (il fonctionne « en boucle ouverte »), il s'agit d'un système dit « déterministe ». Dans notre exemple précédent, un tel programme casserait certainement l'œuf. Cependant, si le programme autorise une réaction qui ajuste les interventions qu'il pilote, le système fonctionne « en boucle fermée » et est dit « stochastique ». Ici, les micro-interrupteurs ou les capteurs servent à définir une pression de prise suffisante pour saisir correctement l'œuf, mais pas assez forte pour risquer de le briser.

De nombreux robots sophistiqués comportent des capteurs multiples pour mesurer l'intensité lumineuse, la chaleur et d'autres données variables. Ces capteurs peuvent être utilisés pour suivre ce qui se passe pendant que le bras effectue une tâche et pour signaler tout problème éventuel : un robot soudeur qui perce joyeusement des trous sur son propre socle, par exemple. Cela peut arriver !

Mécanisme de transmission

Des courroies dentelées en caoutchouc permettent d'obtenir une démultiplication qui autorise des positionnements d'une précision de l'ordre de 2 mm.

Épaule

Cette articulation a une liberté de mouvement de 180°.

Moteur pas à pas

Tous les mouvements du bras sont produits par un moteur pas à pas qui assure une commande très précise. Chaque fois qu'une impulsion électrique est appliquée, le moteur effectue un pas de rotation — il peut être de 7°.

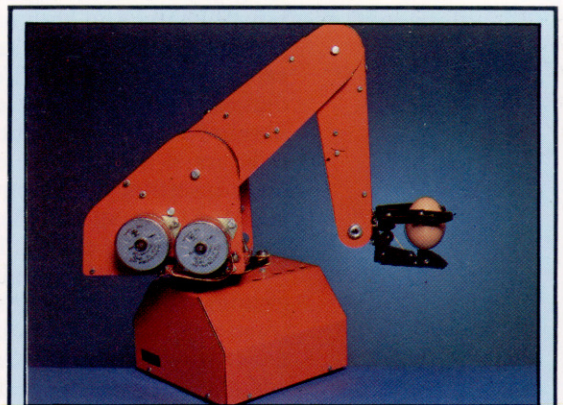
Taille

Le bras peut tourner complètement sur lui-même.

Carte de circuits

Cette carte ne renferme que de simples circuits logiques qui servent à décoder les signaux provenant de l'ordinateur. Il n'y a aucun microprocesseur, ROM ou RAM.

David Weeks



Ian McKinnell

Apprentissage du langage

Il est relativement simple d'écrire un programme qui commande un bras robot. En BASIC, la principale tâche est de permettre à votre ordinateur d'accepter des commandes provenant du clavier et de les transmettre au port du bras à l'aide d'instructions POKE. Similairement, une entrée provenant du bras pourrait être lue à partir du port associé par la fonction PEEK. Si la vitesse est recherchée en premier lieu, il est essentiel d'écrire le programme en langage machine. FORTH est un langage qui offre la facilité de programmation du BASIC et la vitesse associée au code machine. Une opération complète de saisie d'un objet pourrait par exemple être commandée par la simple commande SAISIR. On trouve ce langage sur un grand nombre d'ordinateurs domestiques.

Les fichiers

Après avoir défini une structure globale, nous continuons notre projet en étudiant le problème de la manipulation des fichiers.

Le programme de carnet d'adresses que nous avons développé dans nos derniers cours de programmation BASIC est en fait une version simplifiée d'un programme de base de données, et comme tel il implique la notion de « fichiers ».

En programmation, un « fichier » peut être comparé à des chemises dans un classeur. C'est une collection d'informations stockées ensemble. Les ordinateurs stockent les fichiers sur des bandes ou des disquettes magnétiques. Chaque « fichier » d'informations reçoit un nom unique afin de permettre à l'ordinateur d'y accéder lorsque cela est nécessaire. L'information ainsi stockée peut être un programme ou encore des données utilisées par un programme. Si on prend le carnet d'adresses informatisé comme exemple, l'information nécessaire comprend deux parties distinctes : le programme lui-même, et les données traitées par le programme. Le programme est l'ensemble des instructions qui permettent à l'ordinateur (et à l'utilisateur) de manipuler et de traiter les données.

Les données utilisées par le programme sont celles que vous désirez pouvoir trouver dans un carnet d'adresses — noms, adresses, etc. Mais d'autres données ne sont normalement pas à la portée de l'utilisateur. Il s'agit, ici, des données de gestion interne utilisées par le programme pour faciliter son exécution. Les « drapeaux » font partie de ce type de données; ils peuvent servir à fournir une information relative à la dimension de la base de données (c'est-à-dire le nombre d'enregistrements qu'il renferme), ou à indiquer si un tri a été effectué depuis la dernière insertion d'un nouvel enregistrement, ou encore à indiquer combien de fois un enregistrement particulier a été sollicité ou imprimé.

Dans les premières parties du cours de programmation BASIC, nous avons utilisé les instructions READ et DATA pour intégrer des données dans un programme. Ce genre d'entrée n'est adéquat que lorsque ces données ne risquent pas d'être modifiées, comme le nombre de jours dans un mois. Si leur modification est probable, le programme peut les solliciter; INPUT, INKEY\$ ou d'autres méthodes peuvent être utilisées pour les inscrire dans le programme. Cette méthode pourrait être indiquée pour un jeu où l'on doit deviner un nombre; une portion du programme pourrait se présenter ainsi :

```
PRINT « DEVINER LE NOMBRE »
INPUT N
IF N < > ORDNOB THEN...
```

Les données, dans le programme de carnet d'adresses, sont cependant exposées à de fré-

quents changements. Théoriquement, tous les enregistrements devraient être stockés et lus dans les tableaux appropriés à l'intérieur du programme à l'aide d'instructions READ et DATA. Mais toutes les données devraient alors être entrées à l'intérieur du programme. Si des changements devaient être faits, comme lors de suppressions ou d'insertions de noms et d'adresses, des modifications importantes devraient être apportées au programme lui-même : impression du programme, vérification des modifications nécessaires, écriture de nouvelles portions du programme et entrée de ces éléments dans l'ordinateur. Un plus gros problème interviendrait : les nouveaux segments du programme ne seraient pas des modules de programme complets ayant été testés indépendamment; les modifications seraient alors répercutées au hasard dans tout le programme. La seule façon de savoir si le programme modifié fonctionne correctement serait de l'exécuter.

Heureusement ce n'est pas nécessaire, puisque les données peuvent être stockées indépendamment du programme. On le réalise en créant des fichiers de données sur cassette ou sur disquette. Ces fichiers sont des collections d'enregistrements traités un peu de la même façon que les données dans une instruction DATA.

Les fichiers sont donc des collections de données stockées sur un support de mémoire de masse qui peuvent être utilisées par un ou plusieurs programmes. Un programme de traitement de texte, par exemple, peut accéder au même ensemble de noms et d'adresses pour composer automatiquement des lettres personnalisées.

Les fichiers sont gérés de façon différente selon la version de BASIC utilisée. La meilleure façon de découvrir comment votre ordinateur les gère est de rechercher dans la notice de la machine les explications relatives aux instructions OPEN et CLOSE et d'essayer quelques exemples. La description que nous présentons ici est très généralisée et n'a pour objet que d'illustrer globalement l'utilisation de fichiers. Ceux-ci peuvent être soit à accès séquentiel, soit à accès direct. Dans un fichier à accès séquentiel, l'information est stockée en plaçant d'abord le premier élément d'information suivi de l'élément suivant, et ainsi de suite. Un fichier à accès direct est disposé de telle sorte que l'ordinateur peut localiser directement l'élément de données requis sans avoir à les parcourir toutes depuis le début du fichier.

Un fichier séquentiel pourrait être comparé à un film; vous commencez à regarder le film et

vous suivez son déroulement jusqu'à la fin. En revanche, regarder un film à l'aide d'un magnétoscope est un peu comparable à l'utilisation d'un fichier à accès direct; vous pouvez avancer la bande, la rebobiner ou regarder uniquement les parties désirées. Nous n'examinerons que les fichiers séquentiels parce qu'ils conviennent mieux aux systèmes à cassette.

Supposons que nous désirions conserver une liste des températures moyennes de la semaine de la façon suivante :

LUNDI	13.6
MARDI	9.6
MERCREDI	11.4
JEUDI	10.6
VENDREDI	11.5
SAMEDI	11.1
DIMANCHE	10.9

Pour simplifier les choses, toutes ces données seront traitées comme des données numériques, lundi étant le jour 1 et dimanche le jour 7. Les données deviennent :

1, 13.6, 2, 9.6, 3, 11.4, 4, 10.6, 5, 11.5, 6, 11.1, 7, 10.9

Pour stocker ces données dans un fichier séquentiel, les étapes suivantes doivent être insérées dans le programme :

- OUVRIR (OPEN) le fichier
- Écrire les données dans le fichier
- FERMER (CLOSE) le fichier

Lorsque l'instruction OPEN est utilisée, il est nécessaire d'indiquer si nous désirons écrire des données venant de l'ordinateur dans le fichier (une sortie) ou si nous désirons lire des données dans le fichier pour les transférer dans l'ordinateur (une entrée). Sur le BBC, les instructions OPENOUT et OPENIN sont utilisées. En BASIC Microsoft, les instructions équivalentes sont OPEN « 0 » et OPEN « 1 ». Voici un court programme pour entrer les données ci-dessus dans un fichier (en BASIC Microsoft) :

```
100 OPEN « 0 », # 1, « TEMP. DAT »
110 PRINT # 1, 13.6, 2, 9.6, 3, 11.4, 4, 10.6, 5, 11.5, 6, 11.1,
7, 10.9
120 CLOSE # 1
```

Le mot OPEN de la ligne 100 permet au programme d'accéder au fichier. OPEN est suivi par « 0 » pour indiquer que les données seront envoyées par le programme pour être stockées dans le fichier. Le « # 1 » indique que nous désignerons ce fichier comme le fichier numéro un de notre programme. Chaque fichier reçoit un numéro arbitraire qui sera ultérieurement utilisé avec les instructions INPUT # et PRINT # quand nous désirerons lire ou écrire des données. Finalement, nous plaçons le nom du fichier entre guillemets. Nous avons nommé notre fichier TEMP. DAT pour indiquer qu'il renferme des lectures de température et ne représente pas un programme.

Voici un programme BASIC Microsoft complet qui entre les données dans un fichier, puis les lit et les imprime :

```
100 OPEN « 0 », # 1, « TEMP. DAT »
110 PRINT # 1, 13.6, 2, 9.6, 3, 11.4, 4, 10.6, 5, 11.5, 6, 11.1,
7, 10.9
120 CLOSE # 1
130 REM LES LIGNES 130 ET 140 REPRÉSENTENT
140 REM LES INTERVENTIONS DU PROGRAMME
150 OPEN « 1 », # 1, « TEMP. DAT »
160 FOR X = 1 TO 7
170 INPUT # 1, JOUR, TEMP
180 PRINT « JOUR »; JOUR, TEMP
190 NEXT X
200 CLOSE # 1
210 END
```

Cette opération ouvre un fichier, numéroté # 1 et nommé TEMP. DAT, écrit des données à l'aide de l'instruction PRINT # puis le ferme. Plus tard dans le programme, le même fichier est ouvert en utilisant le numéro et le nom (le numéro n'a pas à être le même qu'à l'origine, mais le nombre utilisé dans les instructions PRINT # ou INPUT # doit être le même que celui affecté au nom de fichier lors de son ouverture). INPUT 1 de la ligne 170 indique que l'entrée proviendra d'un fichier numéroté # 1 (c'est-à-dire le fichier TEMP. DAT) et non à partir du clavier.

Nous devons laisser ce problème de manipulation de fichier un moment. Revenons au programme de carnet d'adresses et à certains des éléments impliqués dans la sous-section « Initialisation » du programme. Voyons d'abord la quantité de mémoire nécessaire pour un seul enregistrement du carnet d'adresses. (Le mot fichier désigne ici un ensemble d'enregistrements associés dans une base de données, et non un groupe de données stockées sur une disquette ou sur une cassette sous un même nom par le système d'exploitation.)

L'utilisation de champs d'une longueur déterminée est un gaspillage d'espace mémoire, mais rend le programme beaucoup plus simple. Si nous affectons une ligne complète pour chaque champ, avec 40 caractères par ligne, l'ensemble sera sauvegardé dans un tableau même si la ligne renferme surtout des espaces. Cependant, dans certaines versions de BASIC, chaque élément d'un tableau de chaîne peut compter jusqu'à 256 caractères. L'instruction de déclaration du tableau (DIM) ne fait que définir le nombre d'éléments et non la dimension de chaque élément.

Si vous avez un BASIC qui peut manipuler des tableaux multidimensionnels, il serait possible d'utiliser une dimension séparée pour chacun des champs. Mais c'est impossible dans de nombreuses versions de BASIC. Nous devons donc rechercher d'autres approches. La méthode la plus simple serait d'utiliser une chaîne distincte pour chacun des champs. Mais il faut « tricher » si vous désirez utiliser des tableaux multidimensionnels même si votre BASIC ne peut les gérer.

L'astuce est de traiter tous les éléments qui devraient être dans le tableau multidimensionnel comme s'ils étaient contenus dans un tableau unidimensionnel. Par exemple, un tableau à deux dimensions de trois lignes et de cinq colonnes serait déclaré ainsi : DIM(3,5) et ren-

I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

fermerait un total de quinze éléments : A(1,1) à A(3,5). La même information pourrait être contenue dans un tableau à une dimension : A(15). Partout où l'on devrait mentionner un élément A(R,C), nous pourrions utiliser à la place l'expression A((R - 1) * 5 + C).

Si nous utilisons un tableau de chaîne distinct pour chaque champ, nous devons décider comment déclarer les tableaux. La manière la plus simple serait d'utiliser une dimension de tableau fixe, mais en limitant le nombre total d'enregistrements que nous pouvons stocker dans la base de données. Une meilleure approche serait de définir la dimension du tableau selon le nombre d'enregistrements utilisés. Certains BASIC limitent la taille des tableaux de chaîne. Même si ce n'est pas le cas de votre version de BASIC, un nombre important d'enregistrements dans la base de données occupera rapidement tout l'espace disponible de la mémoire. De nombreuses versions de BASIC vous permettront cependant d'affecter dans le tableau autant d'éléments que vous le désirez jusqu'au moment où tout l'espace disponible aura été utilisé. Chaque fois que le programme vous demande QUELLE DIMENSION DE TABLEAU?, entrez une plus grande valeur jusqu'à ce que vous obteniez un message d'erreur. L'instruction CLEAR de la ligne 100 sert à éliminer le tableau à la fin de chaque essai. Sans cette instruction, vous obtiendriez un message d'erreur à la ligne 30 en tentant de déclarer une deuxième fois le tableau.

```

10 READ D$
20 INPUT « QUELLE DIMENSION DE TABLEAU? »; A
30 DIM N$(A)
40 FOR L = 1 TO A
50 LET N$(L) = D$
60 NEXT L
70 FOR L = 1 TO A
80 PRINT L, N$(L)
90 NEXT L
100 CLEAR
110 GOTO 10
120 DATA « ABC INFORMATIQUE »
130 END
    
```

Même si 40 caractères sont uniquement autorisés dans chaque élément, avec cinq champs par enregistrement, et si l'on prévoit 256 éléments pour chaque tableau, la quantité de mémoire requise pour contenir toutes ces données dans la mémoire principale devient énorme. Si 1 octet est nécessaire pour stocker chaque caractère, nous aurions alors besoin de 51 200 octets (5 × 40 × 256) uniquement pour les données. Il n'est évidemment pas possible d'utiliser tant de mémoire.

Malheureusement, comme nous l'avons déjà mentionné, les routines de manipulation de fichiers sont un peu difficiles à utiliser. Si nous désirons éviter d'utiliser des fichiers externes, l'autre possibilité de cette alternative serait de mettre les données dans une instruction DATA, de telle sorte qu'elles restent présentes dans le programme. En plus des contraintes que cela impose au niveau de la capacité de la mémoire,

cette technique rend très difficile toute modification des données et accroît le risque d'erreur. Il est donc préférable d'utiliser les fichiers de données externes. Dès que vous aurez essayé quelques courts programmes servant à les manipuler, le processus global deviendra de plus en plus facile à comprendre. Comme illustration, nous avons choisi deux ordinateurs et deux versions de BASIC très différents pour mettre en œuvre notre court programme de température quotidienne écrit en BASIC Microsoft. Il s'agit du Sinclair Spectrum et du BBC. Ces deux versions diffèrent considérablement de notre BASIC Microsoft.

Dans le BASIC du Spectrum, les instructions OPEN et CLOSE sont réservées au Microdrive. Lorsque le stockage sur cassette est utilisé, des versions spéciales des commandes SAVE et LOAD sont nécessaires. La commande ordinaire SAVE sert à stocker des programmes et des variables de programme sur bande (et, évidemment, des données ordinaires dans des instructions DATA). Les tableaux peuvent être sauvegardés à l'aide de l'instruction SAVE-DATA. Voici comment cela se présente :

SAVE nom de fichier DATA nom de tableau (l)

Le nom de fichier est le nom donné au fichier (TEMP.DAT dans le programme Microsoft). Le nom de tableau est simplement celui du tableau de chaîne suivi de parenthèses vides. Pour « sauvegarder » les températures quotidiennes, nous devons d'abord déclarer un tableau de chaîne et y écrire les données, peut-être à l'aide d'instructions READ-DATA. Pour rendre la différence entre le nom de fichier et le nom du tableau plus évidente, nous nommerons le tableau c\$ et le nom de fichier sera « TEMPDAT ».

```

10 DIM c$(14)
20 FOR x = 1 TO 14
30 READ c$(x)
40 NEXT x
50 DATA 1, 13.6, 2, 9.6, 3, 11.4, 4, 10.6, 5, 11.5, 6, 11.1, 7, 10.9
60 SAVE « TEMPDAT » DATA c$(l)
70 STOP
80 LOAD « TEMPDAT » DATA c$(l)
90 FOR L = 1 TO 14 STEP 2
100 PRINT « JOUR »; c$(L), c$(L + 1)
110 NEXT L
120 STOP
    
```

La ligne 60 sauvegarde toutes les données du tableau de chaîne c\$ dans un fichier portant le nom de TEMPDAT. Le programme s'arrêtera à la ligne 70, et vous devez à ce moment rebobiner la bande. La touche CONT relancera le programme. La ligne 80 inverse le processus et charge les données du fichier TEMPDAT dans le tableau c\$.

Le BBC possède un des BASIC les plus sophistiqués parmi les ordinateurs domestiques. Il permet une programmation structurée avec des structures évoluées comme RÉPÉTER JUSQU'À et l'utilisation de « procédures ». Avant de pouvoir en utiliser une, elle doit d'abord être définie. Nous verrons comment l'effectuer dans la version de notre programme pour le BBC.



Notez que le BASIC du BBC définit la direction de la circulation des données dans l'instruction OPEN en utilisant OPENIN ou OPENOUT :

```

10 DIM C$(2,7)
20 FOR L = 1 TO 7
30 FOR C = 1 TO 2
40 READ C$(C,L)
50 NEXT C
60 NEXT L
70 DATA 1, 13.6, 2, 9.6, 3, 11.4, 4, 10.6, 5, 11.5, 6, 11.1, 7, 10.9
80 INPUT « TAPEZ S POUR SAUVEGARDER DES DONNÉES » ;
  K$
90 IF K$ <> « S » THEN GOTO 80
100 PROCsauver : CLEAR : DIM C$(2,7)
110 INPUT « REBOBINEZ LA BANDE DONNÉES ET TAPEZ L » ;
  K$
120 IF K$ <> « L » THEN GOTO 110
130 PROCcharger
140 PRINT « JOUR TEMP »
150 FOR L = 1 TO 7
160 FOR C = 1 TO 2
170 PRINT C$(C,L); «   »
180 NEXT C : PRINT
190 NEXT L
200 END
300 DEF PROCsauver
310 X = OPENOUT (« TEMPDAT »)
320 FOR L = 1 TO 7
330 FOR C = 1 TO 2
340 PRINT # X, C$(C,L)
350 NEXT C
360 NEXT L
370 CLOSE #X
380 ENDPROC
400 DEF PROCcharger
410 X = OPENIN (« TEMPDAT »)
420 FOR L = 1 TO 7
430 FOR C = 1 TO 2
440 INPUT # X, C$(C,L)
450 NEXT C
460 NEXT L
470 CLOSE #X
480 ENDPROC
    
```

Un des avantages du BASIC du BBC est que les instructions de manipulation de fichiers fonctionnent à la fois pour les fichiers sur cassettes et sur disquettes. Les programmes écrits pour cassettes peuvent être utilisés sans aucune modification si un lecteur de disquettes est ajouté par la suite.

Jusqu'ici, nous avons vu comment les données peuvent être transférées à partir d'instructions DATA par l'intermédiaire de tableaux vers des fichiers de données sur bande (ou sur disquette) et vice versa. La prochaine étape examinera le processus d'INITIALISATION afin de voir combien de tableaux seront nécessaires et avec combien d'éléments, et à quels endroits du programme les données devront être transférées vers ou en provenance de ces tableaux.

Exercices

Écrivez un programme en utilisant les éléments suivants :

- DÉBUT
- INITIALISATION
- ENTRÉE DES DONNÉES
- CHOIX
 - Sauvegarde des données
 - Chargement des données
 - Sortie du programme
- FIN

INITIALISATION initialisera toutes les variables et tous les tableaux utilisés par le programme. Les données comprendront quinze noms, entrés au clavier à l'aide de messages de sollicitation affichés à l'écran. CHOIX proposera un menu à l'utilisateur : SAUVEGARDER DES DONNÉES?, CHARGER DES DONNÉES? ou QUITTER LE PROGRAMME?. Essayer de créer un drapeau dans la partie SORTIE DU PROGRAMME qui sauvegardera automatiquement les données si aucune sauvegarde n'a déjà été effectuée.

Variantes de basic



Parmi les ordinateurs domestiques les plus populaires, seul le BBC accepte les noms de variables longs comme NOM\$. Le Spectrum accepte des noms longs de variables numériques, mais uniquement des noms de variables de chaîne à une lettre. Le Dragon 32, le Vic-20 et le Commodore 64 acceptent des noms de variables longs, mais seuls les deux premiers caractères sont significatifs; FNOM\$ est donc valide mais désigne la même variable que FNOMBRE\$ puisque les deux premiers caractères sont identiques. Sur l'Oric-1, les noms de variables ne peuvent compter plus de deux caractères (d'abord une lettre puis une lettre ou un nombre).



Sur le Dragon 32 vous devez utiliser ce format :

```

OPEN « 0 », #1, « TEMPDAT »
PRINT #1, 1, 13.6, 2, 9.6, etc.
CLOSE #1
    
```

et

```

OPEN « 1 », #1, « TEMPDAT »
INPUT #1, D, T
CLOSE #1
    
```



Sur le Commodore 64 et sur le Vic-20 utilisez ce format :

```

OPEN 1, 1, 2, « TEMPDAT »
PRINT #1, 1, 13.6, 2, 9.6, etc.
CLOSE 1
    
```

et

```

OPEN 1, 1, 0 « TEMPDAT »
INPUT #1, D, T
CLOSE 1
    
```



Le Lynx et l'Oric-1, dans leur version standard, n'acceptent pas les fichiers cassette. Cependant, des extensions permettront bientôt la gestion de tels fichiers.



J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

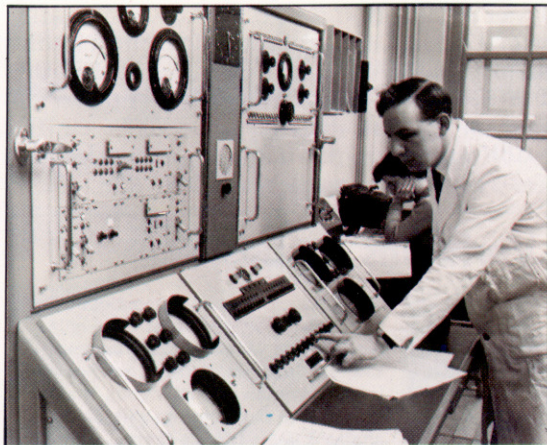


J. Lyons

L'informatique de gestion débuta dans des applications plutôt inattendues.

Gestion électronique

Contrairement aux ordinateurs antérieurs qui étaient conçus pour des applications militaires ou scientifiques, le LEO 1 était conçu seulement pour effectuer des opérations arithmétiques simples impliquant des milliers d'éléments ou de transactions par jour.



En 1947, une société décida de construire un ordinateur pour automatiser le travail de bureau. Cette machine allait être le premier ordinateur de gestion. L'initiative fut prise par J. Lyons, une société de distribution de thé. L'activité de cette entreprise impliquait un grand nombre de petites transactions, et, afin de rentabiliser ce type de commerce, il était primordial d'obtenir le maximum d'efficacité dans les tâches administratives. Même après les dures années de la Seconde Guerre mondiale, la société employait plus de mille agents de bureau.

En raison de ces contraintes, les responsables de Lyons acquièrent rapidement une réputation d'innovateurs : ils introduisirent des machines à calculer dans leurs boutiques dès 1896, et, en

1930, ils expérimentaient déjà l'enregistrement des transactions sur microfilm. Au même moment, ils mirent sur pied un premier centre de recherche pour étudier des méthodes d'exploitation commerciales. Régulièrement, cette société envoyait des représentants à l'étranger afin d'examiner les nouveautés qui pourraient lui être utiles. En 1947, deux employés furent envoyés aux États-Unis pour voir les « nouveaux cerveaux électroniques ». Mais ils ne tardèrent pas à découvrir qu'un ordinateur était construit dans leur propre pays, à Cambridge.

Le conseil d'administration de Lyons commanda une étude devant examiner la construction possible d'un ordinateur. L'étude révéla qu'une telle construction coûterait 1 200 000 F mais permettrait de réaliser des économies de 600 000 F par an. En conséquence, un projet commença en 1947. L'entreprise dut freiner son enthousiasme, car l'ordinateur de Cambridge n'en était encore qu'au stade de la conception. Lyons accorda un montant de 36 000 F à l'université de Cambridge pour l'aider à construire ce qui allait devenir l'EDSAC (*Electronic Delay Storage Automatic Computer*). Ce montant servit à acheter au gouvernement des surplus de tubes. En 1949, EDSAC réussit à exécuter sa première tâche, calculer une table de nombres primaires!

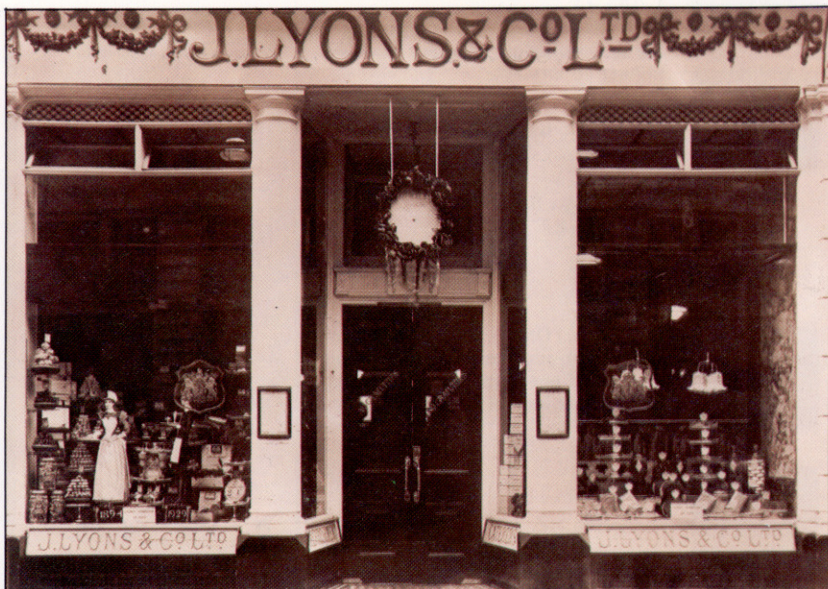
Lyons analysa les problèmes que son ordinateur devait résoudre et élaborer les routines nécessaires. Ces études devinrent les esquisses des premiers programmes et aidèrent à orienter la conception du matériel. Mais il devint rapidement évident qu'un ordinateur de gestion était fondamentalement différent d'un ordinateur de recherche universitaire. Alors que l'EDSAC était conçu pour exécuter des opérations mathématiques longues et complexes, l'ordinateur de gestion devait résoudre un type de problème complètement opposé. Les opérations mathématiques étaient minimales — simplement additionner et multiplier — mais le volume des données à traiter était énorme.

LEO (*Lyons Electronic Office*) devint opérationnel le 9 février 1954 et fut utilisé pour calculer la paie hebdomadaire des mille sept cents membres du personnel. Il pouvait effectuer en une seconde et demie une opération qu'un employé de bureau résolvait en huit minutes.

LEO fut une véritable réussite pour la société Lyons où l'on se rendit rapidement compte qu'il fallait plusieurs machines. D'autres sociétés s'y intéressèrent et Lyons mis sur pied une société de fabrication et de commercialisation d'ordinateurs pour profiter de l'expérience acquise. Leo Computers eut beaucoup de succès et produisit des versions améliorées de LEO qui trouvèrent des débouchés dans l'industrie, dans l'administration et dans la gestion commerciale. La société fut rachetée en 1963 par l'English Electric Company.

Magasin de thé électronique

Les boutiques traditionnelles Lyons offraient un cadre curieux pour accueillir les premières applications sérieuses d'informatique, mais elles représentaient, en fait, le type d'activité commerciale qui se prêtait le mieux à des méthodes de gestion informatisées.



PROGRAMME N° 6

THÈME

Nombres aléatoires et parties entières.
Ces deux notions, importantes pour vos programmations futures, introduisent deux fonctions : fonction RND et fonction INT.
Tapez ce programme :

```
$LIST
15 PRINT RND (1)
20 GOTO 15
25 END
```

```
$RUN
.73278014
.675114126
.18262696
.782880158
.175432536
```

Le RND de la ligne 15 signifie : RaNDom (« aléatoire » en anglais).
La machine, grâce à cette instruction RND, produit des nombres aléatoires décimaux compris entre 0 et 1.
Stoppez le déroulement des résultats en tapant CTRL - C simultanément.
Remplacez maintenant 15 par : 15 ? RND (6).
Tapez RUN.

```
$LIST
15 PRINT RND (6)
20 GOTO 15
25 END
```

```
$RUN
.185497918
.490230353
.4637721
.363803188
.698794455
.317724406
.306142324
.374171723
```

Vous constatez que cela ne change rien.
Transformez de nouveau la ligne 15 en écrivant :

15 ? RND (0)

Exécutez cette ligne.
Stoppez le programme et comparez le nombre inscrit à celui affiché à l'écran.

```
$LIST
15 PRINT RND (0)
20 GOTO 15
25 END
```

```
$RUN
.374171723
.374171723
.374171723
.374171723
.374171723
.374171723
```

(suite)

```
.374171723
.374171723
.374171723
.374171723
.374171723
```

RND (0) donne le dernier nombre aléatoire engendré par le système.

REMARQUES

Les nombres décimaux aléatoires compris entre 0 et 1 ne sont pas très pratiques. Les nombres entre 0 et 9 sont beaucoup plus faciles à utiliser. Voici un programme pour obtenir des nombres entiers aléatoires compris entre 0 et 9.

```
$LIST
10 REM CREATION D'UN NOMBRE ALEATOIRE
```

```
20 Z = RND (1)
30 REM ON MULTIPLIE Z PAR 10
40 Z = Z * 10
50 REM ON ARRONDI
  AU PREMIER NOMBRE ENTIER INFERIEUR
60 Z = INT (Z)
70 PRINT Z
80 END
```


\$RUN	(suite)
0	4
5	1
7	7
6	3
5	3
0	3
2	2
1	1
2	6
6	9
5	4
5	4
7	7

La ligne 60 utilise la fonction INT.
L'instruction INT (Z) produit le plus grand nombre entier inférieur ou égal à Z. Cela correspond à la fonction partie entière. Les parenthèses suivant INT peuvent contenir une expression arithmétique ou variable quelconque. Pour faire engendrer des nombres de 1 à 10 au lieu de 0 à 9, il suffit d'ajouter 1 à Z en écrivant dans le programme précédent :

65 Z = Z + 1

```

$LIST

10 REM CREATION D'UN NOMBRE ALEATOIRE
20 Z = RND (1)
30 REM ON MULTIPLIE Z PAR 10
40 Z = Z * 10
50 REM ON ARRONDI
  AU PREMIER NOMBRE ENTIER INFERIEUR
60 Z = INT (Z)
65 Z = Z + 1
70 PRINT Z
80 GOTO 20

```

\$RUN	(suite)
5	6
3	3
9	5
9	3
2	2
10	8
2	5
10	3
8	9
10	2
5	5

Si ce programme vous paraît trop complexe et si vous voulez le contrôler à chaque étape, modifiez-le comme ceci à l'aide de l'instruction ? et de nombreux REM.

```

$POKE33,33

$LIST

10 REM CREATION D'UN NOMBRE ALEATOIRE
20 Z = RND (1) : PRINT « Z = RND (1) », Z
30 PRINT
40 REM ON MULTIPLIE Z PAR 10
50 Z = Z * 10 : PRINT « Z = Z * 10 », Z
60 PRINT
70 REM ON ARRONDI X A L'ENTIER INFERIEUR
75 Z = INT (Z) : PRINT « Z = INT (Z) », Z
80 PRINT
90 REM ON ADDITIONNE 1 A Z
100 Z = Z + 1 : PRINT « Z = Z + 1 »
110 PRINT Z
120 PRINT
130 REM FAIRE UNE PAUSE
140 FOR I = 1 TO 2000 : NEXT I
150 GOTO 20

```

```

$RUN

Z = RND (1)                .307079968

Z = Z * 10                 3.07079968

Z = INT (Z)                3

Z = Z + 1                  4

Z = RND (1)                .928337745

Z = Z * 10                 9.28337745

Z = INT (Z)                9

Z = Z + 1                  10

```

```

Z = RND (1)                .32139633

Z = Z * 10                 3.2139633

Z = INT (Z)                3

Z = Z + 1                  4

Z = RND (1)                .488417322

Z + Z * 10                 4.88417322

Z = INT (Z)                4

Z = Z + 1                  5

```